

## Exact and Heuristic Methods for Integrated Container Terminal Problems

Iris, Cagatay; Larsen, Allan; Røpke, Stefan; Pacino, Dario

*Publication date:*  
2016

*Document Version*  
Publisher's PDF, also known as Version of record

[Link back to DTU Orbit](#)

*Citation (APA):*  
Iris, C., Larsen, A., Røpke, S., & Pacino, D. (2016). Exact and Heuristic Methods for Integrated Container Terminal Problems. DTU Management Engineering.

## DTU Library

Technical Information Center of Denmark

---

### General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

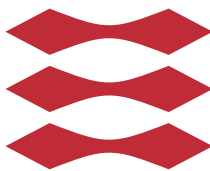
- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

# Exact and Heuristic Methods for Integrated Container Terminal Problems

Çağatay İris

DTU



Kongens Lyngby 2016

Technical University of Denmark  
Department of Management Engineering  
Management Science  
Produktionstorvet, building 426,  
2800 Kongens Lyngby, Denmark  
Phone +45 4525 4800  
[www.man.dtu.dk](http://www.man.dtu.dk)

# Summary (English)

---

The introduction of a standardized logistical unit for all modes of transport has proved a major technological innovation. Since its inception in 1956, containerization has made enormous inroads into the carriage of traditional break bulk cargoes which are now transported on specialized vessels through specialized container terminals. The revolutionary homogenization of heterogeneous cargoes has resulted in massive increases in the size of vessels and terminals, a trend that continues until now. The related developments of globalization, mega container vessels, hub-and-spoke network configurations, increased service frequencies, slow steaming, severe competition between terminals and others create new challenges for the container terminals that need to achieve higher operational efficiency with limited resources. In an environment of scarce financial and spatial resources, improvements in operations planning often present superior cost efficiency in comparison to solutions engaging capacity expansions by new investments.

Planning container terminal operations is not an easy task since it involves multiple interdependent problems. These problems have attracted the attention of many researchers in the field of operational research and have been studied for more than few decades. Traditionally, planning problems are being solved hierarchically. However, hierarchical solutions can be poor, misleading or even infeasible for later stages. Integration of the relevant problems has emerged as a possible approach. This thesis deals with optimization of integrated container terminal problems for seaside and yardside operations. More specifically, it addresses the berth allocation, quay crane assignment, quay crane scheduling, operational stowage planning, transfer vehicle assignment and scheduling problems that modern container terminals are confronted with.

The general goal of the thesis is to increase the modeling, methodological and computational knowledge on a set of integrated container terminal problems. The main contributions are summarized in individual chapters in which problems are formulated as mathematical models and solved by a spectrum of exact and heuristic methods. The study approach starts with integrated seaside operations and expands gradually to cover the integrated seaside-and-yardside problems. Some of the thesis chapters cover state-of-the-art problems and present new solution methods, improvements on the available formulations, and improved computational results, while others focus on novel problems and present new results along with benchmarks generated. The methodological contributions of the thesis cover domains of both deterministic and stochastic problems for integrated container terminal problems. The computational results suggest that the methods constitute improvements with respect to solution quality and computational time. Managerial conclusions for the studied problems are also provided in the relevant chapters of the thesis.

# Preface

---

This thesis was carried out at Technical University of Denmark (DTU) and submitted to the Management Science division, DTU Management Engineering as a partial fulfilment of the requirements for acquiring a Ph.D. in Engineering. Assoc. Prof. Allan Larsen was the main supervisor of the Ph.D. student, while Prof. Stefan Ropke and Assoc. Prof. Dario Pacino acted as co-supervisors.

The title of the thesis is "Exact and Heuristic Methods for Integrated Container Terminal Problems" and it consists of six chapters. The first chapter (1) introduces the domain of container terminals, presents different planning problems and contributions of the thesis. The next two chapters (2, 3) develop exact and heuristic methods for the integrated berth allocation and quay crane assignment problem. Following chapter (4) focuses on berth and quay crane scheduling problem under uncertainty and develops exact methods to solve the problem. Finally last two chapters (5, 6) introduce a new integrated container terminal problem for ship loading operations and develop modeling and solution methods for that problem. The thesis is based on five academic papers that have been either published/under-review or is currently in final stages for submission in international peer-reviewed journals/books. All papers are co-authored.

The project was funded by the Danish Maritime Cluster under the "The Danish Maritime Cluster: a skill development project" title (Danish: Danmarks Maritime Klynge: et maritimt kompetenceudviklingsprojekt). The Ph.D. studies were conducted between February 1st 2013 and May 1th 2016.

Kongens Lyngby, May 5th 2016.

Çağatay İris



# Acknowledgements

---

The path going to a PhD thesis is enjoyable, but not easy. I have obtained great help from everybody that I have met and shared ideas on this way.

Foremost, I would like to thank my advisors Assoc. Prof. Allan Larsen, Prof. Stefan Ropke, and Assoc. Prof. Dario Pacino. They have supported me all the way to become a better researcher and deliver high quality research outcomes. I deeply appreciate their interests in the project and all the constructive comments, suggestions they have made. I would like to open a huge parenthesis for Dario Pacino. His strong knowledge on container terminal operations has opened many new aspects in my research. I have greatly enjoyed his views and encouragements. I regard him as my mentor and most of all as a friend. Stefan Ropke was my light of Eärendil in the research, he was there when all other lights went out (Quote LoTR). Finally I thank Allan Larsen for his leadership to manage my studies in a perfect flow. Every single thing that I have learnt from them will company me in the future.

As the visiting research fellow at National University of Singapore (NUS), I strongly appreciated the supervision of Prof. Der-Horng Lee and the fruitful research collaboration with Asst. Prof. Jian Gang Jin. I would like to thank them both for many interesting ideas and discussions.

I would like to thank all colleagues and friends at DTU Transport and DTU Management Engineering for the support and companionship. I would like to thank APM Terminals - Cargo Service A/S, Aarhus C, especially Martin Laursen for our collaboration.



I dedicate this thesis to my mother Sureyya Iris who has supported my education even under the toughest conditions. I also thank my father Ayhan Iris (ladies first, dad, sorry!), my brother Cagri Iris, my huge family and friends back in Turkey. I also would like to thank my ex-girlfriend Nukte Sandikci for dedication along the way.

Last but not least, I am thankful to DTU for awarding me the PhD scholarship and to the Danish Maritime Cluster (DMC) who has funded the project. Likewise, Augustinus Fonden, Otto Mønsted Fonden, Oticon Fonden, and Frants Allings Legat Fonden have supported my external research stay financially.





# Contents

---

<b>Summary (English)</b>	<b>i</b>
<b>Preface</b>	<b>iii</b>
<b>Acknowledgements</b>	<b>v</b>
<b>1 Introduction and Motivation</b>	<b>1</b>
1.1 Container terminals around the globe . . . . .	4
1.2 Technicalities of container vessels and container terminals . . . . .	6
1.3 Decision problems in container terminals . . . . .	12
1.3.1 Strategic problems in container terminals . . . . .	12
1.3.2 Tactical problems in container terminals . . . . .	14
1.3.3 Operational problems in container terminals . . . . .	15
1.4 Integrated container terminal problems . . . . .	19
1.4.1 Collaborative container terminal problems . . . . .	24
1.5 Research Scope and Contributions . . . . .	25
1.5.1 Thesis Contributions . . . . .	28
1.6 Conclusion . . . . .	33
1.7 Future Work . . . . .	35
1.7.1 Problem Definitions . . . . .	35
1.7.2 Extensions of Methods . . . . .	36
<b>2 Integrated Berth Allocation and Quay Crane Assignment Problem: set partitioning models and computational results</b>	<b>39</b>
2.1 Introduction . . . . .	40
2.2 Literature Review . . . . .	42
2.2.1 BACAP literature - BAP, QCAP properties . . . . .	42
2.2.2 BACAP literature - Objective function properties . . . . .	43
2.2.3 BACAP literature - Solution techniques . . . . .	44

2.3	Modeling the BACAP: Meisel and Bierwirth (2009) Model . . . .	46
2.3.1	Time-variant model . . . . .	49
2.3.2	Time-invariant model . . . . .	52
2.4	Generalized Set Partitioning Formulations . . . . .	53
2.4.1	Time-Invariant GSPP Model . . . . .	53
2.4.2	Time-variant GSPP Model . . . . .	55
2.4.3	Discretization policies . . . . .	58
2.4.4	Column Reduction Strategies and Valid Inequalities for set partitioning models . . . . .	58
2.4.5	Discussion of solution methods . . . . .	64
2.5	Computational Results . . . . .	65
2.5.1	Benchmark instances and running conditions . . . . .	65
2.5.2	Upper bound and warm start strategies . . . . .	66
2.5.3	Computational Results . . . . .	67
2.6	Conclusions and suggestions for future work . . . . .	77
<b>3</b>	<b>Improved formulations and an adaptive large neighborhood search heuristic for the integrated berth allocation and quay crane as- signment problem</b>	<b>79</b>
3.1	Introduction . . . . .	80
3.2	Literature Review . . . . .	82
3.3	Problem Description . . . . .	83
3.4	Mathematical Model: Meisel and Bierwirth (2009) Model . . . .	85
3.4.1	Valid Inequalities and Variable Fixing Methods for BACAP	88
3.5	Adaptive Large Neighborhood Search (ALNS) heuristic for BACAP	95
3.5.1	Construction Heuristic . . . . .	99
3.5.2	Destroy Operators . . . . .	99
3.5.3	Insertion Operators . . . . .	102
3.6	Computational Results . . . . .	107
3.6.1	Data and experimental settings . . . . .	107
3.6.2	Results for improved formulations . . . . .	109
3.6.3	Results for ALNS . . . . .	114
3.7	Conclusions . . . . .	119
<b>4</b>	<b>A two-stage stochastic programming approach to berth and quay crane scheduling problem under uncertainty</b>	<b>121</b>
4.1	Introduction . . . . .	122
4.2	Related literature . . . . .	123
4.3	Problem definition and mathematical formulation . . . . .	125
4.3.1	Master problem and recourse function . . . . .	128
4.3.2	Valid inequalities for master problem . . . . .	133
4.4	Solution method: Integer L-shaped method . . . . .	136
4.4.1	Integer L-shaped method . . . . .	137
4.4.2	Accelerating strategies . . . . .	140

4.4.3	Alternative strategies . . . . .	142
4.5	Computational results . . . . .	143
4.5.1	Data set . . . . .	143
4.5.2	Results . . . . .	144
4.5.3	Risk-averse stochastic programming for BCSP under un- certainty . . . . .	148
4.6	Conclusion and future research . . . . .	151
<b>5</b>	<b>A survey on the Ship Loading Problem</b>	<b>153</b>
5.1	Introduction . . . . .	153
5.2	The Ship Loading Problem . . . . .	155
5.3	Literature Review . . . . .	157
5.3.1	Operative Stowage Planning Literature . . . . .	158
5.3.2	Load Sequencing Problem Literature . . . . .	159
5.3.3	Equipment Assignment and Scheduling Literature . . . . .	161
5.3.4	Integration Efforts . . . . .	162
5.4	Conclusions and Future Research Directions . . . . .	165
<b>6</b>	<b>Formulations for ship loading problem with transfer vehicle as- signment and scheduling</b>	<b>167</b>
6.1	Introduction . . . . .	168
6.2	Relevant literature . . . . .	170
6.3	Ship Loading Problem . . . . .	171
6.4	Mathematical Model for the SLP . . . . .	174
6.4.1	Enhancements for the SLP model . . . . .	177
6.5	New lower and upper bounds for the SLP . . . . .	181
6.6	Computational analysis . . . . .	183
6.6.1	Results . . . . .	184
6.6.2	Hierarchical vs integrated planning . . . . .	188
6.6.3	What is the minimum fleet size? . . . . .	190
6.7	Conclusion and future research direction . . . . .	191
<b>A</b>	<b>Appendix of Chapter 2</b>	<b>193</b>
A.1	Proof of Theorem 1 . . . . .	193
<b>B</b>	<b>Appendix of Chapter 3</b>	<b>195</b>
B.1	Calculation of $\hat{q}$ , $f_1$ , $f_2$ (Iris, Pacino, Ropke, and Larsen (2015b))	195
B.2	Lower bound on objective function (Iris et al. (2015b)) . . . . .	196
B.3	Smarter greedy insertion: pseudo code . . . . .	196
	<b>Bibliography</b>	<b>199</b>



## CHAPTER 1

# Introduction and Motivation

---

In the preface of "The Box" (Levinson, 2008), it is stressed that *the history of the shipping container is humbling. Careful planning and thorough analysis have their place, but they provide little guidance in the face of changes that alter an industry's very fundamentals*. Levinson (2008) notes that containerization is the cutting edge of maritime freight transport history. Bringing a uniform unit for all modes of transport, *containers* ("The Box") helped to integrate all entities in the *shipping network*. This integration also includes *container terminals*<sup>1</sup> which are special type of harbors that are accustomed to handle containers. The general aim of this thesis is to increase the knowledge on container terminal planning problems and supply a better guidance for further applicability of optimization methods to increase the efficiency of operations.

Benefiting from rapid globalization, the container shipping network has been steadily growing. In 2014, a container volume of 170 million Twenty Equivalent Units (TEUs) is expected to be transported around the globe through this network (UNCTAD, 2015). In the following years, the global container throughput is expected to have a growth rate of over 6% and it is reported that this value is outperforming world Gross Domestic Product (GDP) growth continuously (UNCTAD, 2015). This shows that there is a modal shift from bulk to con-

---

<sup>1</sup>Container terminals and terminals will be used interchangeably in this thesis



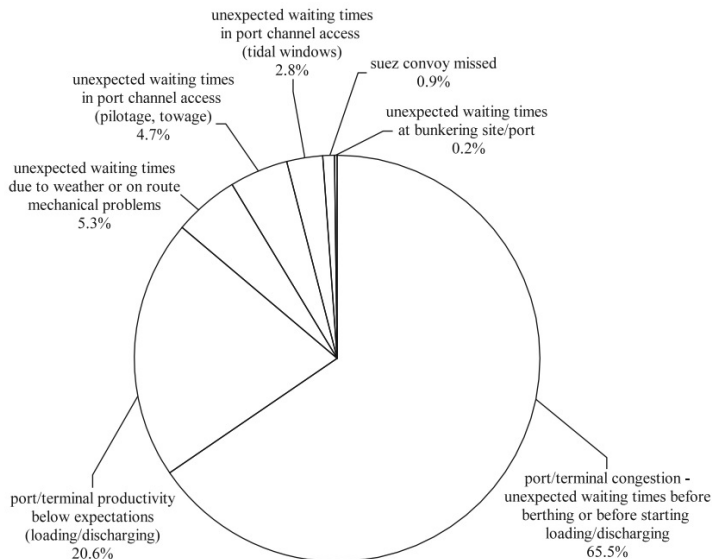
tainerization which means that a wider range of goods is placed in containers, and transported through container terminals consequently. The same report points out that leading container terminals handle more than 30 million TEUs annually, therefore we can note the increasing volumes as one of the factors that makes operations planning more complex and important for container terminals.

Liner shipping companies have adapted to this trend by increasing the capacity of their fleet, deploying *container vessels* of over 19.000 TEUs. Capacity is, however, not enough. A reliable service requires the goods to arrive on time, and it is here where container terminals face another challenge with the increasing vessel sizes. First, the infrastructure of container terminals is under strong pressure since larger vessels require longer berths with deeper water depths and more equipment for loading/unloading operations. Second, the larger vessels create tremendous peak of workloads at the terminals. Summing up, the increase in vessel capacity is another factor that makes the efficient planning more complex and important for container terminals.

It is not only the size of the vessels is increasing, they also sail slower in the open sea (Psaraftis and Kontovas, 2010). Note that the design speed of the recently-born mega-vessels is around 19 knots which is slower than their predecessors (See Triple-E guidelines as an example). This points out that the shipping industry intends to keep the *slow-steaming* principle. There are two major consequences of slow-steaming for container terminals. First and foremost, considering the short *transit time* promise of the liner shipping companies with more times spent in the open sea, there is a clear expectation on the reduction of the time spent at container terminals for each container. Second, liner shipping companies, or alliances that they form, operate at higher frequencies to compensate the capacity losses due to slow-steaming. This results in more frequent vessel visits which means heavier workloads for the terminals.

The importance of container terminals in the entire shipping network can be better understood with Figure 1.1. According to Notteboom (2006), a survey data (which is made in 2004) show the sources of unreliability in the major East-West container transport trade. Survey results show that over 85% of causes for unreliability is due to port systems. The biggest portion is the congestion and resulting waiting times before/after loading/unloading at ports (65.5%). The second biggest portion is poor port productivity (20.6%). Such data support that the efficiency expectations of liner shipping companies are not perfectly met by port operators.

The need for the efficient planning is not only imposed by customers. There is an intense competition between container terminals. The biggest four terminal operators (Hutchison Port Holdings (Hong Kong), PSA International (Singapore), DP World (United Arab Emirates), APM Terminals (the Netherlands))



**Figure 1.1:** Sources of liner schedule unreliability (survey data of East Asia – Europe relations, Notteboom (2006))

control the 26.7% of all World traffic. The competition between top tier terminals is more intense. Jiang, Chew, and Lee (2015) stress that terminals of the same region compete to attract more cargo and gain a higher share of the market. It is noted that the cargo flow shifts in Asian ports are more influenced by this competition (Jiang et al., 2015). Competitive advantage can be achieved by making investments on terminal resources such as extending the berthing and storage area, increasing the number of equipment or renovating the current equipment, etc. Investments, however, are at a very high cost and there is no guarantee of an improvement in the service quality because there is a limit to the number of equipment that can be deployed to a vessel and inefficient management of these new investments can bring more congestion and deterioration in the overall performance. In this sense, increasing the efficiency without any investment on the infrastructure gives a serious competitive advantage in the market.

As a key node in the maritime transport network, container terminals are in active business with the other entities in the network. A container terminal, as an entity, has a "system" and an "environment". The system corresponds to the controllable container terminal business involving many complex operations which will be detailed in the following sections. A large proportion of the content in this thesis deal with problems arising in the container terminal

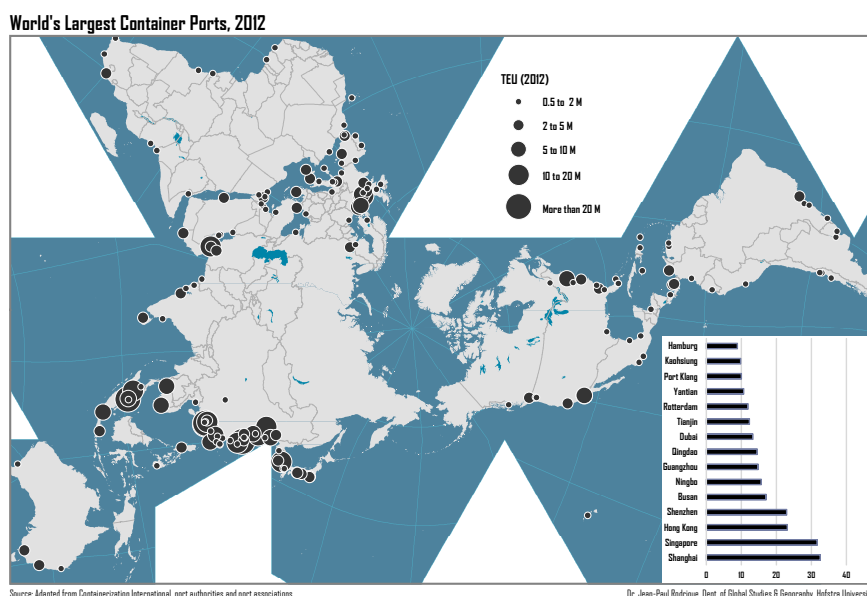
system. The environment is the exogenous system that affects operations and cannot be controlled by the container terminal. The environment (such as the information flow between liner shipping company and terminal, weather, vessel arrival times, etc.) brings a degree of uncertainty into operations planning. This makes efficient planning an even more complicated task. This thesis will also address problems arising from this uncertainty.

Considering all of these challenging prospects while doing the required business, container terminals aim at utilizing all resources and coordinating activities in the best possible way at the same time. Practitioners, researchers, and developers of decision support tools have been devoted to solve these challenges. These efforts seek cost-efficient, reliable and robust solutions to various problems of the container terminals. With respect to research papers focusing on these problems, there is an increasing trend in the number of papers (See reviews on Operations Research (OR) methods to solve container terminal problems; Vis and de Koster (2003), Steenken, Voß, and Stahlbock (2004), Stahlbock and Voß (2007), Kim and Lee (2015)). All related studies regarding thesis problems will be detailed in the following sections.

Methods presented in this thesis can be updated to adapt to new situations including operational or tactical changes such as varying vessel arrival schedules, equipment renewal with respect to numbers or technicalities, new safety restrictions, new operational rules, etc. They can provide better planning for critical resources in the terminal. A better cost calculation will be supplied for higher quality input to contract planning with customers. The details of contributions will be explained later in this chapter. The thesis focuses on new methods and/or improvements on existing methods which span through a vast spectrum of OR literature. The methods developed in this thesis might also help for building more efficient decision support tools for container terminals.

## 1.1 Container terminals around the globe

In order to get into details of operations, the container terminals around the World are reviewed briefly. The top 20 container terminals accounted for approximately 45.7% of world container port throughput in 2014 (UNCTAD, 2015). These terminals showed a 4.5% increase in throughput compared to 2013, the same as the estimated increase for 2013. Figure 1.2 illustrates container terminals with different handling volumes in the World with respect to 2012 statistics. Figure 1.2 shows that many of the leading container terminals are in developing economies, many of which are in Asia, the remaining few big terminals are in developed countries in Europe and North America.



**Figure 1.2:** World's leading container terminals Rodrigue (2013)

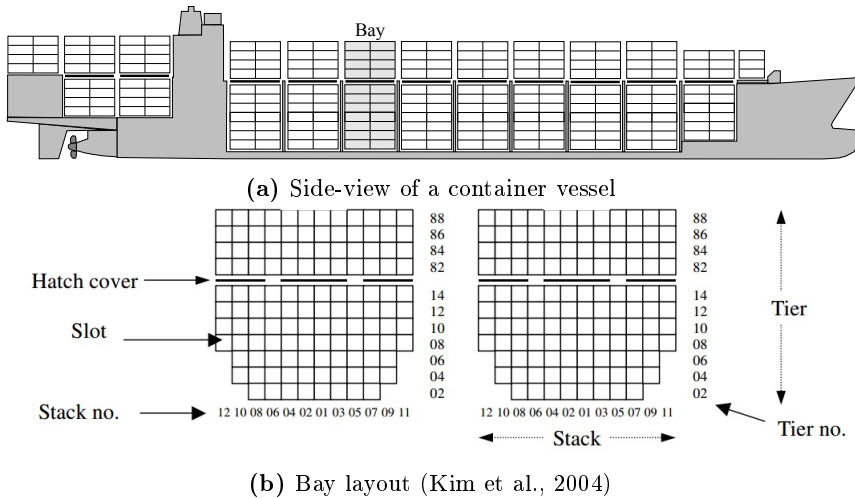
Container terminals should be evaluated not only with container handling volumes but also by the hinterlands they are connected to and their target business models. Signifying the importance of Asian and Western European manufacturing zones, the leading positions of the terminals in these regions can be easily understood. Traditionally these terminals mainly handle *import and export containers*. Export containers are the ones which are stowed aboard a vessel for export, while import containers are unloaded from the vessel and transported to hinterland area. Terminals in Asian provinces (like Shenzhen, Ningbo, Qingdao, etc.) mostly load full export containers to Western Europe, while unload empty/less-value-loaded containers. Regarding the target business models, the *hub-and-spoke* system has been initiated in shipping networks where mega container vessels visit a limited number of transshipment terminals (hubs) while small vessels (feeders) connect the remaining relatively small terminals (spokes) in that region to respective hubs. Container terminals like Singapore rises as the most popular transshipment terminal with a transshipment percentage of more than 80% of the overall handling volume.

Such trends do not only affect the leading container terminals on the main ocean routes. Relatively small terminals are also influenced by the increase in the trade volumes, larger vessel sizes and hub-and-spoke system. As a result, all container

terminals should focus on improving operations and try to stay competitive in the market. The productivity of a container terminal does not only determine its competitiveness and attractiveness, it also affects the entire network in which the terminal is involved. Thus the efficient operations planning should be the aim of any types of terminal.

## 1.2 Technicalities of container vessels and container terminals <sup>2</sup>

Container vessels are ships that are designed for transportation of the containers. Nowadays, container terminals are *called* by numerous container vessels. The classification of these vessels is usually made according to their capacity and size. We refer reader to Pacino (2012) for different vessel and container types. The layout of a container vessel is presented in Figure 1.3. Containers are (un)loaded from/to positions (slots) of the vessel. Figure 1.3a illustrates the bays (holds) which are ordered through the entire vessel length. Each bay has a number of stacks and tiers. Figure 1.3b illustrates the bay-layout where tiers are vertical indexing of positions in each bay, while stacks (rows) are horizontal indexing of positions through the bays of a vessel. There are hatch covers which are metallic blocks and they separate containers on-deck and below-deck.



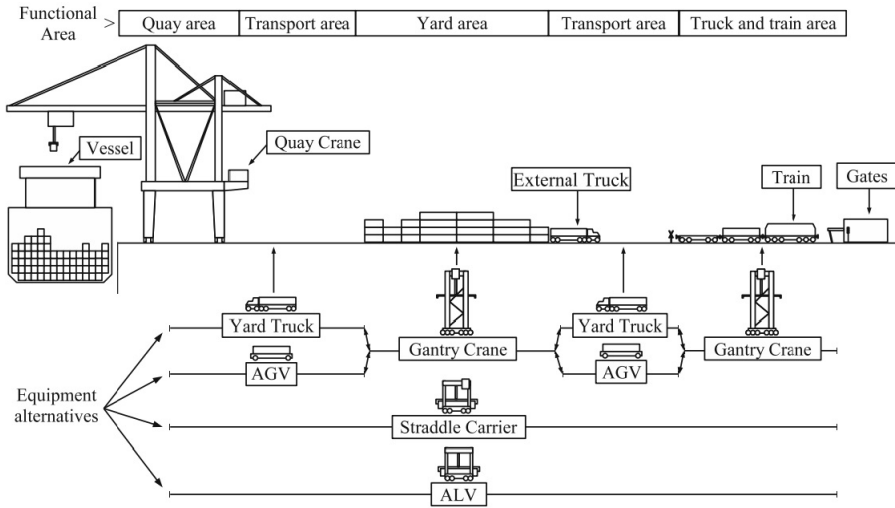
**Figure 1.3:** A container vessel

<sup>2</sup>Inspired by Meisel (2009a); Kemme (2013); Carlo, Vis, and Roodbergen (2014a,b)

A container terminal has been divided mainly into four operation areas in terms of functionality to serve the vessels.

- Quayside (seaside) area: the area where vessels are berthed and containers are (un)loaded from/to the vessel
- Transport area: the area where containers are transported from/to the vessel
- Yard area: the area where containers are stored for inbound or outbound transport
- Truck, train and gate area: the borders of terminal where containers are transported in/out of the system

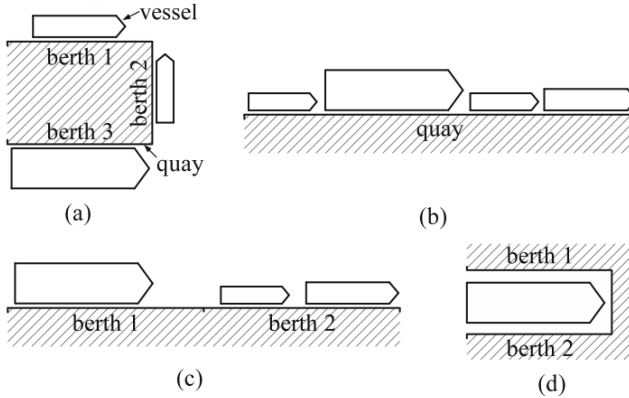
Figure 1.4 illustrates these functional areas and equipment alternatives that can be used in these areas (Meisel, 2009b).



**Figure 1.4:** Schematic side view of a container terminal (Meisel, 2009b)

The quayside (seaside) area is composed of the *berth* where vessels moor, and the (un)loading equipment. The berth is the interface between the vessel and the container terminal. Berths with different lengths and configurations are used in terminals. The physical berth configuration is mostly decided in the construction stage of container terminals. With respect to the physical layout, there are mainly three types of berths namely discrete, continuous and hybrid. In discrete layout, the quay is partitioned into a number of berths in which only one vessel can be fit (berthed) at the same time. This type of partitioning can either be due to the construction of the quay (Figure 1.5a) or it can be a decision by the authorities. In continuous layout, a vessel can berth at any point

along the quay (Figure 1.5b). This is mostly a single shore line around 0.5-2 km of length. In hybrid layout, the terminal uses discrete and continuous types together (Figure 1.5c). A special version of hybrid layout is intended berths in which two opposing berths exist, and they can be used simultaneously to handle a vessel (Figure 1.5d).



**Figure 1.5:** Different berth layouts (Meisel, 2009c)

Terminals operate a number of equipment with different characteristics. The very details of each equipment will not be explained in the thesis. We refer the reader to Stahlbock and Voß (2007), Meisel (2009b), Bose (2011) for further explanation for different equipment types. Instead the most popular equipment types will be explained.

Regarding the equipment used in seaside operations, *Quay Cranes* (QCs) so called Ship-to-Shore gantry cranes (STS cranes) are mostly used for (un)loading operations of containers onto(from) vessels. QCs of various size and specialties are used in modern terminals and the QC technologies have been improving continuously. (See further information on QCs in Stahlbock and Voß (2007), Kemme (2013)). Depending on their length, vessels may be served by up to six QCs simultaneously. Recently container terminals mostly use rail-mounted gantry QCs which move in a rail-mounted system. For this reason, QCs cannot overtake each other. In order to deploy a QC to a vessel, all other QCs, which are between the berthing position of the vessel and the respective QC, must be moved. There are also mobile harbor cranes which are more flexible with respect to movements. Rail-mounted gantry QCs are, however, more efficient compared to them (Kemme, 2013). By being faster, rail-mounted gantry QCs can make more moves per hour. QCs also differ in their reach, one of the biggest super Post-Panamax QCs has an outreach of 25 stacks across the vessel bays.

A QC starts loading a container by locking it with the spreader. Then it moves the container to the required vessel position (slot) and it is unlocked, there are twistlocks under each container that help to lock containers to each other. Some vessels are equipped with cell guides which help to position containers. Meanwhile, a gang starts the lashing operations on a group of containers with respect to the lashing pattern. After QC loads the container, it moves back to quay and picks the next container. There are some technical improvements so that the handling speed of QCs is increased. Some QCs are equipped with two trolleys and are called Dual Trolley QCs. One of the trolleys works on the quayside and the other loads the container on the vessel. There is a platform in the middle where each container is being exchanged. Another advancement in QC operations is the dual-cycling (Goodchild and Daganzo, 2006). In this mode of operation, after a QC loads a container from the quay to the vessel, instead of returning directly to the quay, it unloads another container from the vessel onto the quay. This doubles the number of containers carried during a QC cycle.

Containers are transported from/to vessel to/from yard through the transport area by using various equipment (transfer vehicles). These equipment are categorized as passive or active depending whether the equipment has a container-lifting device (Kempe, 2013). As active transport equipment (that can lift up containers from the yard), Straddle Carriers (SCs), forklift trucks, reachstackers and Automated Lifting Vehicles (ALVs) are extensively used, while popular passive transport equipment are terminal trucks (chassis systems), multi trailers and Automated Guided Vehicles (AGVs) (Wiese, Suhl, and Kliever, 2011). Active transport equipment help to decouple QC and transport operations. With the use of active transport equipment, a QC can (un)load a container without the involvement of the transport vehicle and the transport vehicle does not have to wait for the QC to drop the container. This setting might require container buffer areas (Vis, de Koster, and Savelsbergh, 2005) or additional lanes under the QC. The quayside area in front of QC is divided into multiple lanes. There are usually four lanes in front of the QC, two of them for container pickup/drop, one for placing the gearbox and one spare for safety reasons. Additionally, active transport equipment mostly travel into yard blocks to pickup the containers. For this reason, transport times for each container are higher compared to passive transport equipment (Carlo et al., 2014a).

A comparison between automated and manned equipment is also relevant. With respect to required driver resource, AGVs and ALVs which are used as transport equipment in highly automated terminals, are unmanned equipment, while SCs, forklift trucks, reachstackers, terminal trucks require a driver on board. For unmanned equipment, the technological advances to prevent deadlock and congestion are actively used. However, such transport equipment are more vulnerable to disruptions (failure, breakdown, control problem, etc.) due to highly



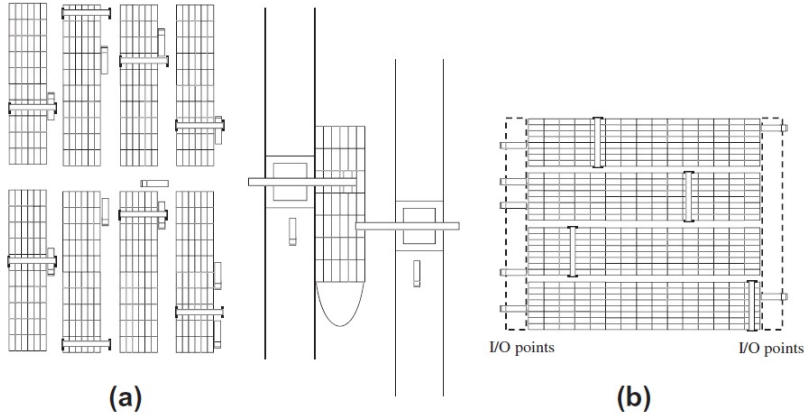
automated characteristics. (See Vis (2006a) for a literature survey on planning problems of AGV-based terminals). There is also a distinction between the use of the driving lanes of the yard. Automated vehicles require to use pre-determined routes to travel and their moving direction is mostly decided beforehand, while manned vehicles are more free to use the area and this gives a higher flexibility of movement.

Descriptions given thusfar cover a single terminal. Recently many container terminals around the World contain multiple terminals to serve the vessels. Especially with the increasing volumes of transshipment, two related vessels can be berthed at different terminals. Therefore there could be a significant need for inter-terminal transportation which should be conducted by the container terminal (Lee, Jin, and Chen, 2012). Such transport operations are mostly performed by yard trucks.

The yard area is composed of the yard in which containers are temporarily stacked and yard equipment to handle the containers in the yard. A yard usually has an import/export container stock area, empty container stock area(s) and refrigerated (i.e. reefer) container area(s). Reefer containers require plugs for electricity needs, while empty containers require service operations (washing, repair, etc.). Hence these containers are mostly separated from the others. In the yard, there can also be some facilities (sheds) for operations such as repair, packing, controlling, etc. A typical yard layout consists of multiple rectangular blocks. Each yard block consists of a number of bays; and each bay has several rows. Containers can be stacked in vertical tiers mostly up to five-seven containers heights. The layout is similar to the configuration of a vessel presented in Figure 1.3.

There are two dominating equipment types that are used for stacking; the gantry (yard) crane (GC) and Straddle Carriers (SC). GCs are used for stacking and moving containers to input/output (I/O) point, i.e., where the transfer vehicle and the yard crane exchange containers. SCs pick up the container directly from the yard and transport it to the QC. Terminals, that use GCs, have a yard area which is more compact, blocks are very close to each other and there is no driving lane in the blocks. This layout type is called block stack (Brinkmann, 2011), another layout type is linear stack in which SCs can enter in the yard block to pickup or drop containers and tiers are at most 4-container heights because of the technical properties of the SCs (Brinkmann, 2011). There are two main yard configurations for the block stack layout and they are presented in Figure 1.6. One of them is parallel (Asian), the other is perpendicular (European) depending on the alignment of containers with respect to the quay. In Figure 1.6a, passive transport vehicles enter the lanes between blocks and they wait in front of the bay that required container will be picked up (Indirect Transfer System) while GC travels to bay that container is stacked and moves the container to(from)

the passive transport vehicle. In Figure 1.6b, I/O points are located at both ends of the blocks (Carlo et al., 2014b). Most commonly AGVs are used to (un)load containers on the seaside, while transfer trucks are used for gate area (Direct Transfer System). A detailed comparison of two layouts can be found in Carlo et al. (2014b).



**Figure 1.6:** Different yard layouts for block stack (Carlo et al., 2014b)

There are two types of GCs, rubber-tyred gantry cranes (RTGCs) and rail-mounted gantry cranes (RMGCs). RTGCs can move within and between blocks which gives a higher flexibility for stacking, but rotating the RTGCs is time-consuming process which should be planned efficiently. On the contrary, RMGCs are embedded into the rails and can only work in one block. There are normally multiple RMGCs which work in each block. Efficient planning of stacking operations can be achieved by cooperative scheduling of multiple GCs in the same block. There are two types of GC configurations, namely passing and non-passing. The passing GCs configuration has multiple GCs of different heights that allows GCs to overtake each other. The non-passing configuration does not allow GCs to pass over each other. The cooperative planning of stacking can increase the efficiency of operations in both settings (Carlo et al., 2014b).

Wiese et al. (2011) have shown that RTGCs are used in 63.2% of all terminals, mainly in Asian continent, while SCs are used in 20.2% of all terminals, mainly in European ports. The remaining equipment for stacking and transport are sparse, RMGCs are used in around 8.8% of terminals studied. With respect to new technologies, ALVs, which are active versions of AGVs, are in use. They can perform the transport to the transfer point where container is exchanged to the yard block. The latest trend in SC is the twin-load technology which is used in DP World Southampton.

Finally there is the truck, train and gate area (a part of the landside) which links the terminal with the hinterland operations. Arriving trucks should first obtain a clearance in the gate where containers are checked for documentation and current loads. Recently many container terminals manage arrival of trucks with booking systems. If SCs are in use, there is an area where SCs can pick up containers from the trucks.

## 1.3 Decision problems in container terminals

Container terminals are complex logistics systems that have many operations and these operations are interacting due to the continuous container and information flow between them. When a vessel arrives at the terminal, it is moored to a berthing position, meanwhile QCs, which will work on that vessel, are mobilized to this berthing position. The quayside is dedicated for (un)loading operations. Then QCs (un)load containers in a given order, while required containers should be retrieved from/to the quay aligned with this order. Vehicles transport the required export containers to respective QCs that will load that container to given position. A similar flow is observed for import containers with a reverse directions of operations. Here the yard functions as the temporary storage area for these containers. Decision problems correspond to the managerial decisions about these operations.

The planning of operations (i.e. solution to a decision problem) is usually made in advance. Each problem has a planning horizon which is the length of time the problem covers ahead. The planning horizon strongly depends on the problem characteristics and it is very important to understand the range of planning horizon in order to make efficient planning. With respect to the length of planning horizon, decision problems can be classified into strategic, tactical, and operational planning problems. Figure 1.7 illustrates various decision problems of each class and direct link between problems.

### 1.3.1 Strategic problems in container terminals

Strategic problems are mostly related to the design features of the terminal, such as the berth, yard layout or location, the selection of equipment types, etc. The decisions made for these problems are very costly and time-consuming to change. The planning horizon of these problems is considered to be the lifespan of the investment. Mostly trade lane economics and regional market demands define a typical solution. Some of these problems will be explained briefly:

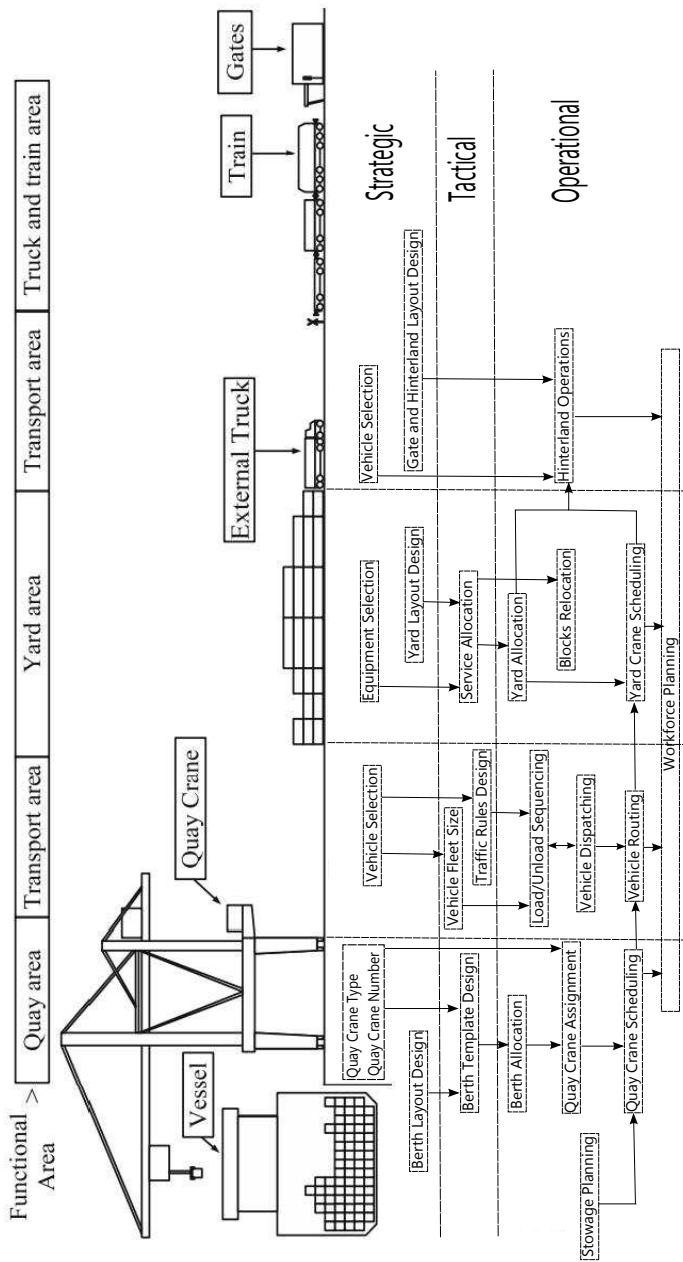


Figure 1.7: Decision problems

**Design of the yard layout:** The problem deals with determining the layout type (parallel or perpendicular), the outline of the yard, the number of aisles between blocks/bays and the size of blocks (de Castillo and Daganzo, 1993).

**Selection of equipment:** The problem deals with determining the specific equipment investments and the technical properties of these equipment. The solutions to these problems also determines the automation level of the terminal (Vis, 2006b; Carteni and de Luca, 2012).

### 1.3.2 Tactical problems in container terminals

Tactical problems consider a planning horizon of months, or at least many weeks. These problems deal with aspects which are not easy to change such as storage management (e.g. allocation of empty container areas), maintenance plans, yard and berth template design, increasing/decreasing vehicle fleet size, etc. Some of the problems will now be detailed:

**Fleet Size Problems:** These problems deal with determining number of transport equipment in the terminals. Depending on required overall workload, terminals can determine required number of fleet (Vis et al., 2005; Vis, de Koster, Roodbergen, and Peeters, 2001). The primary objective of studies in this field is the minimization of the fleet size.

**Service Allocation Problem (SAP):** In this problem, a service is referred to a liner shipping string that has a visit (port call) to the studied terminal. Such strings are mostly cyclic and vessels of the string visit the port frequently. The service allocation problem deals with dedicating specific blocks of the yard for each service (Cordeau, Gaudioso, Laporte, and Moccia (2007)). The objective is to minimize the yard reorganization (housekeeping) which is reflected by the number of reshuffles. This problem is also called yard template design problem. A variant of the SAP focuses on container types (export, import, transshipment, empty and reefer containers) for each vessel rather than different liner shipping strings.

**Berth Template Design Problem:** The motivation of this problem is similar to the SAP. The main difference is the focus on seaside operations rather than the yardside. Berth template has been generated to find a set of berth-windows for cyclic calling vessels so as to maximize the service objective (Moorthy and Teo, 2006). This problem is also called tactical berth scheduling problem. Imai, Yamakawa, and Huang (2014) extend this problem into a more strategic content in which the terminal chooses vessels to be served (has the flexibility not to choose some) and determines berth template design for selected vessels.

### 1.3.3 Operational problems in container terminals

Operational planning problems seek higher utilization of key resources such as the quay, QCs, transport equipment, yard space and equipment. Most of these problems are highly constrained due to the industrial settings. The majority of the problems aim at minimizing the dwell time of containers in the terminal and/or minimizing the cost of operations including equipment use, labor, etc. The planning horizon for operational problems differs from days to seconds. Details of each problem will be explained:

**Stowage Planning Problem (SPP):** This problem determines the exact position (slot) of each export container in the vessel by considering various stability measures (such as trim, draft, metacentric height, etc.) and physical properties (supporting containers from below, IMO restrictions, lashing patterns, line of sight, etc.) of the vessel. Traditionally the SPP is solved by liner shipping companies because each vessel visits multiple terminals along a string and stability must be ensured for each port visit. What is more, the traditional objective of this problem includes minimizing overstowage which can be obtained by considering the entire string (Ambrosino, Sciomachen, and Tanfani, 2004; Pacino, Delgado, Jensen, and Bebbington, 2011). There are two types of overstowage namely stack and hatch overstowage. A container X overstocks a container Y in a stack, if X is stored above Y and the discharge port of X is after the one of Y, such that X must be removed in order to unload Y (stack overstockage). If container Y is under the hatch, all containers over the hatch must be removed (Pacino et al., 2011). There are a few studies that consider the SPP as the complete responsibility of the terminal (Imai, Sasaki, Nishimura, and Papadimitriou, 2006). Recently a variant of the SPP has been suggested as a container terminal problem by considering the flexibility of specific container assignment with respect to the class-based stowage plans (Monaco, Sammarra, and Sorrentino, 2014). The literature on the SPP will be detailed in Chapter 5.

**Berth Allocation Problem (BAP):** The BAP deals with determining the berthing position (along the quay) and the berthing start time for each vessel in the planning horizon with respect to physical and operational constraints. Physical constraints require that all vessels must be berthed within the boundaries of the quay, and different vessels cannot be at the same berthing position at the same time (Monaco and Sammarra, 2007). There are various operational constraints for the problem such as; all vessels must be berthed and processed within the planning horizon; the availability of some berthing positions might differ due to time windows (Cordeau, Laporte, Legato, and Moccia, 2005); different equipment might be of use (Umang, Bierlaire, and Vacca, 2013); there could be some priorities assigned to each vessel (Imai, Nishimura, and Papadimitriou, 2003) or some of the vessels can have favorite berthing areas determined

with berth template problem (Giallombardo, Moccia, Salani, and Vacca, 2010), etc. In order to solve such a problem, the information about the vessel's length, draft, the expected time of arrival, and the projected handling time should be known a priori. The BAP also considers different quay partitioning types namely discrete (Buhrkal, Zuglian, Ropke, Larsen, and Lusby, 2011), continuous (Imai, Sun, Nishimura, and Papadimitriou, 2005), hybrid layouts (Imai, Nishimura, Hattori, and Papadimitriou, 2007). Considering that many vessels constantly arrive and leave the terminals, the restriction on the berthing times are also studied in the literature. The static BAP assumes all vessels are available to berth at the start of the planning horizon (Dai, Lin, Moorthy, and Teo, 2008), while dynamic BAP imposes vessel arrival times as the earliest berthing start time (Imai, Nishimura, and Papadimitriou, 2001). The goal of BAP is to provide fast and reliable services to vessels. This is reflected in the literature by various objective functions such as minimization of the sum of the waiting and handling times of vessels, the workload of terminal resources and the number of vessels rejected to be served at a terminal, etc. The literature on the BAP and variants will be detailed in Chapter 2 and 3.

**Quay Crane Assignment Problem (QCAP):** The QCAP determines the number of QCs and/or the specific QCs to work on each vessel. The QCAP can be solved to the vessel and/or bay level. Each vessel has technical properties such as length (number of bays), width, stability considerations (Ursavas, 2016) that are relevant for the QCAP. The minimum number of QCs that can serve a vessel simultaneously is a function of its length, and the reach span of QCs might affect its technical availability for some larger vessels. In addition to technical properties, the number of containers to be (un)loaded in the terminal and the expected departure time of the vessel directly affect the number of QCs assigned to the vessel. Another operational property is the contracts between liner and terminals, these contracts ensure a number of moves per hour for each vessel and this sets a lower bound on the number of QCs assigned to the vessel (Legato, Gulli, and Trunfio, 2008). Considering the fact that QCs are mounted in rail, they cannot overtake each other. For this reason, the index of QCs assigned to a vessel should be consecutive (Park and Kim, 2003). By determining specific QCs to serve each vessel, movements of QCs can be arranged for all vessels that are berthed simultaneously. Such movements mostly referred as the crane setup which is a common objective to minimize in the QCAP (Turkogulari, Taskin, Aras, and Altinel, 2016). The literature on the QCAP and variants will be detailed in Chapter 2, 3 and 4.

**Quay Crane Scheduling Problem (QCSP):** Given a QC assignment for the vessel, the QCSP determines a schedule of a set of tasks. There are various precedence relations among tasks and these precedence relations are mostly imposed by the stowage plan. Each position must be handled once by one QC and a QC can handle one position at a time. The complete work schedule of

a QC holds the starting time of (un)loading for each position and the order of (un)loading between positions. A solution to QCSP holds the work schedule of all QCs assigned to that vessel, it avoids overtaking of QCs, and it ensures safety measures of bay difference between consecutive QCs. The term task has various interpretations in the literature. It can be related to the handling of groups of containers within a bay, all containers within a bay, or all containers within a connected area of bays (See Meisel and Bierwirth (2011), Bierwirth and Meisel (2014) for papers of each task definition). Various operational and physical restrictions such as time window of operations (Meisel, 2011), moving times between bays (Legato, Trunfio, and Meisel, 2012), ready times (Moccia, Cordeau, Gaudioso, and Laporte, 2006), scheduling in intended berths (Vis and Anholt, 2010; Chen, Lee, and Cao, 2011) are also studied. The most common objective function of the QCSP is minimizing the makespan of operations.

**Loading/Unloading Sequencing Problem:** After a work schedule, which holds the sequence of ship-bays that each QC will work, is made, the sequence of (un)loading specific containers is determined in this problem. In the other words, this problem deals with determining the order of positions that each QC will (un)load. This problem is highly constrained due to physical positions of the slots (Kim et al., 2004). It is noted that solving this problem for outbound containers is harder compared to inbound containers due to requirements of the stowage plan (Kim and Lee, 2015). This problem can be considered as a variant of the disaggregated QCSP where each task is defined by a single container. The literature on the loading and unloading sequencing will be detailed in Chapter 5.

**Vehicle Dispatching and Routing Problem (VDRP):** The management of transport vehicles in the yard can improve the efficiency of a container terminal significantly. Considering the fact that average cycle time for one QC is around 40 seconds, vehicles should ensure that the QC is never idle. The yard is mostly large enough that transfer time of containers will be measured by minutes. For this reason, efficient dispatching and routing of vehicles should ensure a higher utilization for (un)loading operations. There are two fundamental problems; vehicle dispatching (assignment) determines which vehicle will transport each specific (or a group of) container(s) (Bish, Chen, Leong, Nelson, Ng, and Simchi-Levi, 2005) while vehicle routing determines the paths to be taken, and the related scheduling of each vehicle to pick up and drop a container (Kim and Kim, 1999b,c). Vehicle dispatching problem can be formulated in a static or dynamic environment like the BAP. The static version sees the containers as a set of available resource to handle, while the dynamic version is more of a real-time problem in which the ready time of each vehicle and container is a constraint on the problem. The dispatching alternatives can be enlarged by pooling all vehicles for each QC. This means that a vehicle can serve to different QCs in consecutive orders (Bose, Reinert, Steenken, and Voß, 2000). Vehicle routing problems are most commonly considered together with dispatching problems



(See Stahlbock and Voß (2008) and Carlo et al. (2014a) for extensive reviews). Advances in vehicle technologies such as twin loads (Grunow, Günther, and Lehmann), dual-cycling (Nishimura, Imai, and Papadimitriou, 2005) are actively studied in the VDRP. The common objective functions are minimization of (un)loading time (both makespan and/or travel times), idle time of both QCs and vehicles, lateness, etc.

**Storage (Yard) Allocation Problem (SYAP):** The SYAP deals with the assignment of yard storage space to individual containers (or a group of containers) for temporary storage until the next rehandle. There are differences in the SYAP with respect to container types such as export (Kim, Park, and Ryu, 2000), import (Sauri and Martin, 2011), transshipment (Lee, Chew, Tan, and Han, 2006) because each container type has different characteristics. The export containers should mainly be stacked with respect to a stowage plan, expected loading sequence and the results of the SAP (Chen, 1999). In the contrary, import containers arrive in batches and the pickup times for the hinterland transport are usually unknown so probabilistic modeling (Sauri and Martin, 2011) and segregation methods (Kim and Kim, 1999a) are more common. The consideration of the details of the storage location is also diverse in the literature. SYAP studies focus on yard block (Zhang, Liu, Wah Wan, Murty, and Linn, 2003; Moccia, Cordeau, Monaco, and Sammarra, 2009), sub-block (Jiang, Lee, Chew, Han, and Tan, 2012), bay (Lee, Jin, and Jiang, 2011) or individual slot (Kang, Ryu, and Kim, 2006) levels. Both hierarchical and integrated approaches are studied with respect to details of storage location, the hierarchical approach first focuses on a block then it is followed by the specific slot within that block (Park, Choe, Kim, and Ryu, 2011). In some of the problem definitions, not only the container locations but also individual containers are grouped. The most common grouping method for containers is with respect to specific vessels (Lee and Jin, 2013). The common objective function of such problems includes minimization of the storage yard operations cycle time that includes the time to store, retrieve, and reshuffle containers.

**Yard Crane Deployment/Scheduling Problem (YCDP, YCSP):** Considering that there are multiple yard cranes (YCs) available to serve a yard block area, the YCDP deals with assigning YCs to yard areas with respect to workloads. The YCDP mostly considers RTGCs which can move to different blocks of the yard (Zhang, Wah Wan, Liu, and Linn, 2002; Cheung, Li, and Lin, 2002). The YCSP schedules the pickup and stacking operations for each YC. The YCSP (i.e. routing of GCs) is mostly specific for yard layout types (parallel or perpendicular), the number of YCs in each bay, various passing properties of YCs, etc. (see Carlo et al. (2014b) and Luo, Wu, Halldorsson, and Song (2011) for extensive reviews on stacking operations). Many operational problems such as different ready times (Ng and Mak, 2005), safety distance requirements (Wu, Li, Petering, Goh, and de Souza, 2015) are addressed in the YCSP literature.

The common objectives of the YCDP and YCSP are the minimization of the total delays of YC workload, the average waiting time of equipment.

**Blocks Relocation Problem (BRP) / Reshuffling Problems:** Stacking containers on top of each other is an efficient way of utilizing the limited area of the yard. In this setting, containers on top of each tier are available for direct retrieval. However, due to the unpredictable (un)loading sequences or different vessel orders, yard equipment usually require to retrieve containers that are buried beneath other containers (See Lehnfeld and Knust (2014) for an extensive review on stacking problems). To reach the containers in lower tiers, many containers on top of them should be reshuffled. This reshuffling problem can be addressed before the loading operations start (namely pre-marshalling; Lee and Hsu (2007)) or along with the operations (namely BRP; Jovanovic and Voß (2014)). The main aim of these problems is to reduce the number of unproductive moves to achieve the reshuffling operations.

**Terminal Hinterland Operation Problems:** Import/Export containers arrive/leave the terminal through the hinterland which uses trucks or trains to connect the terminal to the inland region. In the case of a rail system, a variant of the VDRP should be solved to bring containers to trains. The interface with trucks is managed through the gate. Scheduling gate operations with respect to truck arrivals (Chen, Govindan, and Yang, 2013a) and evaluating impact of truck schedules on stacking operations (Asperen, Borgman, and Dekker, 2011) are problems that have been addressed in the literature. Another important aspect is about the modeling and comparison of different train loading policies with varying the storage area strategies (Ambrosino, Caballini, and Siri, 2013). The common objective of hinterland problem is to minimize the waiting times for yard transport and stacking equipment with respect to the hinterland operations.

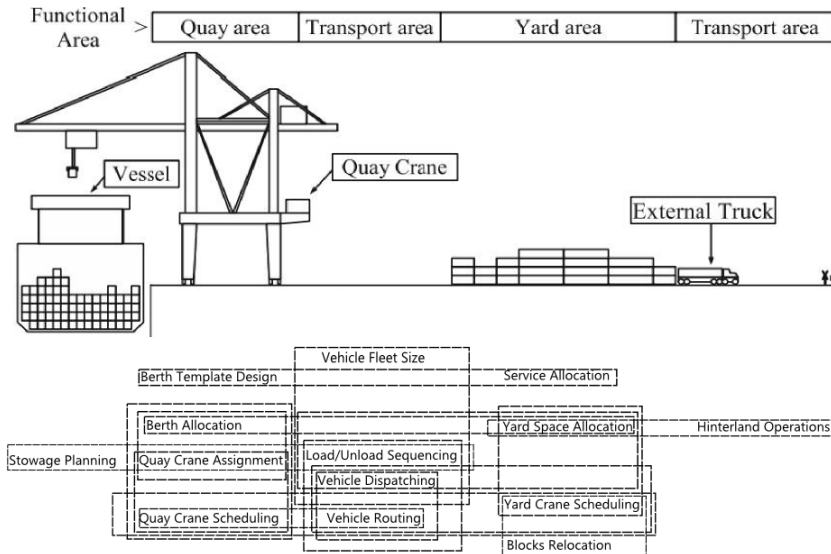
## 1.4 Integrated container terminal problems

The problems described in the previous section are traditionally solved hierarchically by the researchers and practitioners. In the hierarchical planning approach, the solution of a problem is an input for the next stage problem. This relationship follows the direction of physical container flow which is discussed in Section 1.3. In hierarchical planning, the solution of a decision problem can be misleading, poor or even infeasible for the subsequent problems since the planners do not consider the interdependencies between decision problems. In order to overcome such obstacles ad-hoc modifications of plans are made during their executions (Meisel, 2009d). These modifications might reduce the reliability

of terminal services and thus, reduce the satisfaction of the customers (Meisel, 2009d).

There are strong motivations to integrate these problems and solve them as one problem. One of them is that many of these problems share common resources, integrating relevant problems will ensure a higher utilization of the limited resources. It will also bring a better information flow to the next distinct problem so that uncertainties will be relatively reduced for the next stage problems. Planners can also formulate more realistic cost tradeoffs by integrating different problems with different cost components. Since terminals aim at a cost reduction in the overall system, integrating relevant problems will help planners to have a better understanding of the cost structure. It should be noted that integrated problems are harder to solve and it is not easy to reflect all the details of each problem.

Various decision problems spanning from seaside to yardside are integrated in the literature. Figure 1.8 illustrates the integrated problems in a rectangle with subproblems involved. It shows that there are many integrated problems with overlapping subproblems and most of the integrated problems are still operational problems. These integrated problems are now briefly described with their motivations, the different aspects of the integration and the commonly used objective functions. Integrated problems mostly inherit decision variables of the problems that are integrated.



**Figure 1.8:** Various integrated problems

**Berth and Yard Template Design:** The tactical decisions related to berth and yard management can be integrated regarding the relationship that vessels which are berthed with a deviation from the berth template will require more handling in the yard. In the contrary, a differentiation from the yard template will make the berthing time of a vessel longer since the transportation times from yard to vessel will change. Zhen, Chew, and Lee (2011a) consider the violation of vessels' expected turnaround time intervals and the yard cost. These two problems are then solved in an integrated approach. Lee and Jin (2013) focus on this integrated problem and adopt a proactive strategy by designing the calling schedule of the feeders with respect to yard template. Robenek, Umang, Bierlaire, and Ropke (2014) focus on bulk ports where yard locations and berths are specialized for different cargo types with special equipment. The integrated problem becomes significantly important since there is direct a link between equipment type of the yard and the berth.

**Berth Allocation and quay Crane Assignment Problem (BACAP):** The number of QCs assigned to a vessel has a direct impact on the processing time which is an important input for the BAP. In parallel, the berthing position will restrict the movement of remaining QCs which are not assigned to that vessel. Hence integrating these two problems has a strong contribution to the planning. This integration is not only applicable for operational problems of the BAP and QCAP. The tactical berth allocation problem (Giallombardo et al., 2010) covers the assignment of berthing positions and QCs to shipping lines for the longer planning horizons. Chapter 2, 3 review studies in the literature for this problem.

**Berth Allocation and quay Crane Assignment and Scheduling Problem (BACASP):** Incorporating the QC scheduling into the above mentioned integrated problem will help to obtain a better estimation of the processing time of each vessel and a more accurate planning of QC operations (i.e. a better estimation of the setup times and non-crossing requirements). Meisel and Bierwirth (2013) integrate these problems in an interactive hierarchical fashion. Rodriguez-Molins, Salido, and Barber (2014b) focus on the BACASP with the detailed scheduling of the holds where authors aim at minimizing total waiting time. Recently, Turkogulari et al. (2016) also consider a variant of the BACASP where authors formulate a deep integration problem and they solve it to optimality by using a decomposition method. The traditional objective function of such a problem includes time-dependent cost components such as lateness, earliness, makespan, etc. and QC operation costs. Chapter 4 presents a review of papers relevant to this problem.

**Quay Crane Assignment and Scheduling Problem (QCASP):** The number of QCs to work for the vessel(s) (a component of the QCAP) directly affects the work schedule of each QC (i.e. the QCSP). Depending on the available

number of QCs in the berth, scheduling QCs will give a better working time estimate which helps for a better QC assignment. The problem definition covers a single-vessel (Al-Dhaheri, Jebali, and Diabat, 2016) or multiple-vessel planning (Diabat and Theodorou, 2014). In some of the papers, the QCAP is transformed into a QC-to-bay assignment problem hence QC scheduling is incorporated (Theodorou and Diabat, 2014). Many operational constraints such as; safety margins between QCs (Unsal and Oguz, 2013), vessel priority (Fu, Diabat, and Tsai, 2014), vessel stability (Al-Dhaheri et al., 2016) are discussed in the context of the QCASP.

**Quay Crane Scheduling and Vehicle Dispatching/Scheduling Problem:** The QCSP uses the ready times of containers in front of the QC which will load them. These ready times are a function of transport vehicle schedules which are decided by the VDSP. In the unloading process, efficient solving of the VDRP is vital for QC scheduling, because a QC has to drop an unloaded container on a passive transport equipment before it can continue its work (Kaveshgar and Huynh, 2015; Cao, Shi, and Lee, 2010a). Chen, Langevin, and Lu (2013b) consider both loading and unloading operations and they assume that vehicles are shared among different vessels, so that an efficient routing of vehicles prevents starving of the QCs. Tang, Zhao, and Liu (2014) also consider both (un)loading operations so trucks may go back to yardside with inbound containers. A common objective is to minimize the makespan of the operations with a given set of vessels, another objective is minimizing the delay of operations or the idle time of the equipment.

**Quay Crane Scheduling and Yard Management Problem:** The QCSP problem can be integrated to the yard storage problems. Considering that the yard stacks containers of multiple vessels, the congestion of YCs and transport vehicles will affect the ready time of containers. Wang and Kim (2009) note that each block has a limited number of moves that YCs in that block can perform per unit time. This limit and overall load of each YC are studied as a bound on the QCSP. Integrating these problem will control the number of blocks which are assigned to each specific QC because the yard section, which is dedicated, can go beyond the schedule. Recently He, Huang, Yan, and Wang (2015) focus on energy consumption aspects of this integrated problem including the YC scheduling. The common objectives are combination of various cost components such as QC makespan, the total traveling distance of YCs, the expected delay time from interference, and the uniformity of workload among blocks. Another rich QCSP is introduced by Choo, Klabjan, and Simchi-Levi (2010) where authors consider yard congestion in the QCSP, the problem definition is more like a consideration rather than an integration.

**Load/Unload Sequencing and Vehicle Dispatching/Scheduling Problem:** The loading sequence directly affects the transfer time of each vehicle

because containers can be scattered over a wide area in the yard so that planner should make an efficient route to load them all (i.e. the VDRP). In parallel, the ready times of each container in the loading/discharging list determine which vehicles would be available for the pickup and delivery operations (See Jung and Kim (2006) and Shin and Lee (2012) for applications of this problem). Common objectives include the minimization of makespan of operations, idle time of vehicles, etc. Chapter 5 presents a literature review of papers which are relevant to this problem.

**Vehicle Dispatching/Scheduling and Storage Yard Allocation Problem:** Optimizing the storage location to match a particular transfer schedule will offer improvements in reducing the idle times of vehicles, while optimizing the vehicle dispatching for alternative storage allocation will give a higher flexibility to determine a better storage yard allocation. Bish, Leong, Li, Ng, and Simchi-Levi (2001) introduce such a problem for import container with the aim of minimizing the unloading time (See a variant for import containers Lee, Cao, and Shi (2009b) and an extension of both (un)loading operations Lee, Cao, Shi, and Chen (2009a)). Kozan and Preston (2006) focus on the transfer of export containers from the storage area to the vessel, while Bish (2003) integrate scheduling of QCs with assumption that a set of vehicles is pooled among vessels.

**Vehicle Dispatching/Scheduling and Yard Crane Scheduling Problem:** For loading operations of export containers, the synchronization of vehicle scheduling and YC scheduling will reduce the transfer time of all containers and it will also reduce the idle time of equipment (Cao, Lee, Chen, and Shi, 2010b). Li, Goh, Wu, Petering, de Souza, and Wu (2012) consider vehicle waiting times in the YC scheduling problem, this problem, however, does not present an integration method.

**Storage (Yard) Allocation and Yard Crane Deployment/Scheduling Problem:** The workload of a YC is a function of storage allocation decisions made for each container type, so efficient deployment of YCs can be achieved by obtaining a better understanding of optimal storage space allocation. Han, Lee, Chew, and Tan (2008) focus on high-low workload balancing protocol with the objective of minimizing the YC operating costs by designing a non-shared sub-block allocation plan with respect to specific containers. The integration can be further improved by dynamically assigning sub-blocks to incoming containers (Jin, Lee, and Cao, 2016), while Won, Zhang, and Kim (2011) study this problem with/without allowing YC movements between blocks.

### 1.4.1 Collaborative container terminal problems

Containers terminals are in active business with liner shipping companies that operate the vessels, and trucking companies that transport the containers from/to container terminal. An interesting research avenue is expanding to the collaborative problems which integrate decision problems of terminals and their partners. The literature in these problems are less in numbers compared to integrated container terminal problems. Each collaborative problem holds decisions regarding the terminal and the other party.

Collaborative problems with liner shipping companies are mainly related to seaside operations of the terminal. The tactical problems such as berth template design, service allocation, and integration of these problems are collaborative problems since they consider the priorities of the shipping liner and adjust the available terminal resources with respect to these requirements. Recently Wang, Liu, and Qu (2015) improve the level of collaboration for the existing tactical berth allocation problem by proposing two new collaborative mechanisms which are based on the utilities associated with the operations start days of each liner string and inventory cost of transshipment containers.

The collaboration between two parties can also be achieved by integrating the BAP with the preference of ship arrival times. Whenever the decisions about vessel arrival time is incorporated into the BAP and variants, the collaboration realizes. In the literature, there are different ways of setting this relationship. Golias, Saharidis, Boile, Theofanis, and Ierapetrito (2009) consider the amount of emissions produced hourly by each vessel in idle mode for berthing and they plan the vessel arrivals accordingly. Meisel and Bierwirth (2009) impose a cost of earliness if the vessel arrives earlier than an Expected Arrival Time (EAT). This is a reflection of the speedup cost for the liner shipping company. Another way is to integrate the speed optimization problem of the vessels to the BAP. The speed optimization problem (Reinhardt, Plum, Pisinger, Sigurd, and Vial, 2016) determines the sailing speed of the vessel in a leg and it affects both the bunker consumption and the duration of the sailing in the leg. The bunker cost can then be easily incorporated into the BAP objectives. Golias, Boile, Theofanis, and Efstathiou (2010) incorporate the bunker cost for all vessels in transit to their next port of call, while Du, Chen, Quan, Long, and Fung (2011) focus on the leg from their current positions to the terminal for which the BAP is solved (See Wang, Meng, and Liu (2013) for an improved approximation that can handle general fuel consumption more efficiently). Alvarez, Longva, and Engebretsen (2010) propose new berthing policies with respect to the uncertainty of operations and the speed optimization. Recently Venturini, Iris, and Larsen (2016) focus on solving the complete BAP with speed optimization problem between all legs of port visits where authors assume a string holds a

sequence of known port calls.

Another collaboration between terminal and shipping company can be achieved by integrating the BAP with ship routing and scheduling problem. Pang, Xu, and Li (2011) solve tramp shipping routing problem by considering berthing time clash avoidance with different vessels in a given terminal. In a later study, authors consider a deep integrated version of this problem with the transshipment possibility (Pang and Liu, 2014).

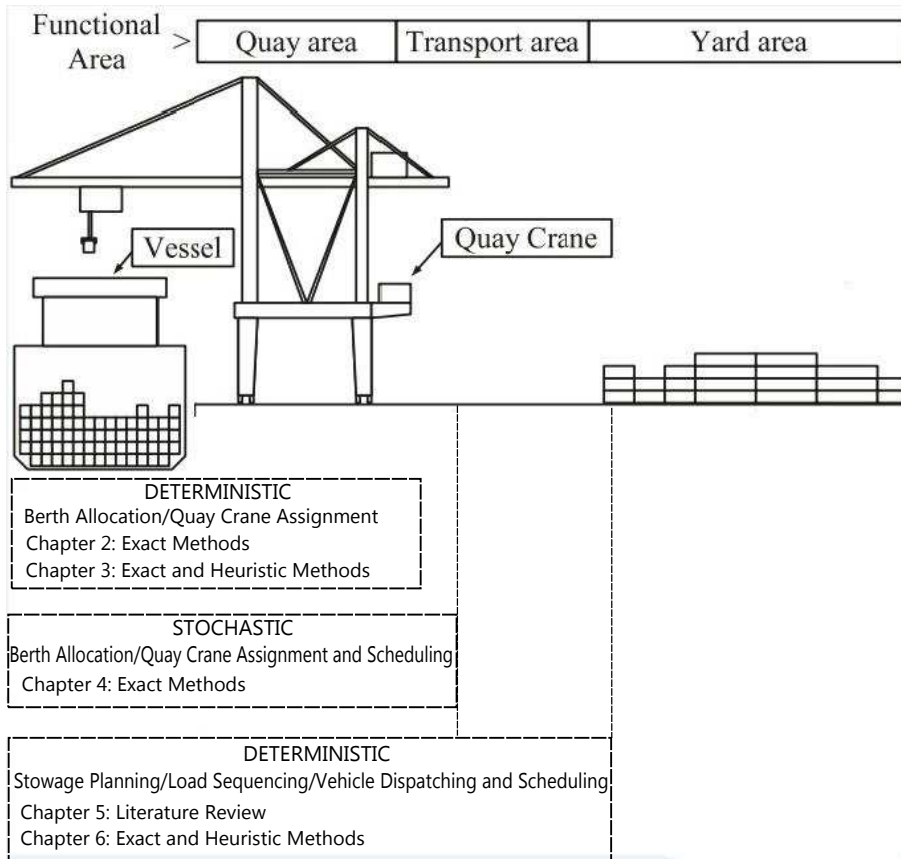
The other interface of the terminal is with truck companies through gates of the terminals. Gate congestion problems can be solved by integrating decisions about the truck arrivals. Recently, there are many studies that focus on gate management problems. However, the number of studies that focus on collaborative planning problem is still limited. Phan and Kim (2016) solve subproblems of determining the optimal schedules for trucks and estimating the expected truck waiting times. These problems are solved iteratively with an improved feed-back fashion. Phan and Kim (2015) have a negotiation framework for the appointment system. This framework allows each trucking company to make its decision on the application of appointment time.

## 1.5 Research Scope and Contributions

This thesis aims at increasing the knowledge on various sets of integrated container terminal problems. It follows the promising research trend to develop new models and algorithms to the integrated problems. It starts with the quayside operations then it investigates possible integration between quayside and yardside problems. All of the studied problems are operational decision making problems. The organizational structure of the thesis is illustrated by Figure 1.9 where each rectangle refers to variants of integrated problems with coverage of functional area(s) and problem settings. It is also shown in this figure that the first three chapters are about the quayside problems, while the last two chapters are about quayside and yardside problems.

The integration of quayside problems started with the involvement of the most important quayside resources namely the berth and QCs. Park and Kim (2003) presented the first problem which was about simultaneous allocation of vessels to the berthing area and determining the number of QCs to work on each vessel (i.e. the BACAP), along with the scheduling of these QCs with the results of the BACAP (i.e. a variant of BACASP). Since then the BACAP has been proven to be an important problem which obtains significant cost savings compared to hierarchical solutions of its subproblems (Vacca, Salani, and Bierlaire, 2013).





**Figure 1.9:** A schematic diagram of thesis organization

Many researchers have focused on different aspects of the BACAP and published important findings. However, the number of studies that focuses on the exact methods to solve the BACAP is still limited. In this thesis, the Chapter 2 in Figure 1.9 focuses on variants of the BACAP and proposes generalized set partitioning problem (GSPP) formulations to solve the problem. The GSPP formulations are built on the well-performing models for the BAP by Buhrkal et al. (2011). The existing formulations on the BACAP are also improved with modeling enhancements which will be detailed in Chapter 3. Considering that there is a limited time for solving the BACAP, a heuristic method which solves the problem efficiently in short computational time is also required, such a method will be detailed in Chapter 3. These two chapters in general aim at increasing the theoretical knowledge on the BACAP and its variants.

The thesis is built in a way that the integration enlarges step by step by including one problem at each step. Looking at the Figure 1.9, the assignment of specific QCs and scheduling of these QCs in a primitive fashion are integrated into a variant of the BACAP in the next step (i.e. a variant of BACASP). Such an addition makes the problem harder to solve mainly due to the non-overtaking requirements of QCs, but that makes it more realistic because terminals have to plan the physical movement of QCs along with their assignments in daily operations (Liu, Wan, and Wang, 2006). The BACASP has mainly been studied with the deterministic inputs. However these problems are strongly influenced by the vessel arrival and processing times which are unknown parameters by the time of the planning. Hence analyzing the BACASP will be more valuable if the uncertainty in inputs is also considered. In Chapter 4, not only the integration is enlarged but also the decision making under uncertainty is visited. Regarding the novelty of the problem and formulations, we have chosen to come up with exact methods for this problem.

Finally, we investigate potentials for innovative integration of terminal problems. This is achieved by taking the integration beyond the seaside operations and including the yardside transport decisions as shown in Figure 1.9. It is known that optimization of load sequencing along with transport vehicle dispatching and scheduling is an integrated problem which has proven its positive contribution (Kim et al., 2004; Jung and Kim, 2006). With this thesis, the innovative aspect materializes by integrating stowage planning into this problem. In particular, we wish to utilize the flexibility that exists while determining detailed operative stowage plan with respect to a given class-based stowage plan which is supplied by the liner shipping company. The flexibility is that liner shipping company supplies a stowage plan with container classes for each slot. The terminal can then decide the stowage plan by means of specific containers and determine the loading sequence along with transport vehicle assignment and scheduling. This integration is studied in the last rectangle in Figure 1.9. Due to the novelty of the problem definition, we first review the literature related to this problem in Chapter 5. Then novel formulations for the problem are presented in Chapter 6.

This thesis attempts to solve these integrated container terminal problems by using a span of OR methods. Some of the methods are exact which aim at finding optimal solutions for the studied problem otherwise present strong upper and lower bounds, while some others are heuristic methods that try to obtain an efficient solution in a relatively rapid fashion. As examples of exact solution methods, a variant of Benders decomposition (Laporte and Louveaux, 1993) and black-box solvers for the Integer linear Programming (IP) formulations (IBM, 2015) are used, while a variant of a large neighborhood search heuristic is used as the heuristic method (Ropke and Pisinger, 2006). The problems of the BACAP and BACASP usually have multiple days of planning horizon, while the time

unit is mostly measured by hours. The problems related to assignment and scheduling of vehicles mostly limited hours with minutes as the unit of time. Therefore, the selection of the solution method and/or running time limits of these methods are inspired by the planning horizon of each problem.

With this thesis, we aim at increasing the methodological and computational knowledge about container terminal problems, however, the question still remains how well container terminals can utilize such valuable information. This thesis inherits observations from the collaborating container terminals in order to formulate new problems or test new methods for some problems. The presented work cannot be used by terminals directly, but some of the knowledge can be added to the decision support tools for planners, and it might help them make better final plans.

### 1.5.1 Thesis Contributions

The thesis is funded by the project "The Danish Maritime Cluster (DMC) – a skill development project" (Danish: Danmarks Maritime Klynge (DMK) – et maritimt kompetenceudviklingsprojekt). The contributions of the thesis are both methodological and computational. In the following, major contributions of the thesis are first listed, then the remaining chapters are detailed where the contributions along with an overview of the dissemination are discussed for each chapter.

The thesis has the following major contributions:

1. Improved knowledge (new best upper and lower bounds, formulations, properties, heuristics, etc.) on the state-of-the-art integrated container terminal problems
2. Optimization of integrated container terminal problems under input uncertainty and analysis of the effects of such uncertainties in the decision making
3. A new integrated container terminal problem which helps to obtain cost reduction compared to hierarchical solutions
4. Flexible solution methods which can solve similar problems with intermediate modifications

The outline of the thesis with detailed contributions of each chapter are now explained.

Chapter 1, **Introduction and Motivation**, of this thesis motivates the problem at hand and reports the background of the container terminal operations. An extensive list of integrated problems is presented along with the motivation of integration and significant state-of-the-art studies. It also reviews the collaborative container terminal problems which are interesting research avenues for future studies.

Chapter 2, **Integrated Berth Allocation and Quay Crane Assignment Problem: set partitioning models and computational results** presents novel set partitioning formulations and various column reduction methods for variants of the BACAP. The literature about the OR methods on the BACAP is quite diverse. We study variants of a well-acknowledged problem (Meisel and Bierwirth, 2009) and compare our results with the available benchmark. We believe that this chapter is also a positive step towards the convergence of problem definitions of the BACAP in the literature. Computational results show that the proposed models significantly improve the best upper and lower bounds of the current state-of-art optimal approaches. We also present a property which shows that minimum cost QC assignment can be achieved by means of a combination of some QC numbers. This property might help to obtain a useful QC assignment in a rapid fashion and it can be of use for future studies. The flexibility of the formulations and the performance of reduction methods are also discussed for similar problems. We show that the model is very effective for the classical BAP with a fine discretization of the berthing space. In this chapter, we also address different QC assignment policies namely time-variant and time-invariant depending on the allowance to change the number of QCs assigned to a vessel when the vessel is berthed. We show that there is an additional cost of time-invariant QC assignment policy and we quantify this difference. The work has been disseminated as follows:

- C. Iris, D. Pacino, S. Ropke, A. Larsen, "Integrated Berth Allocation and Quay Crane Assignment Problem: Set partitioning models and computational results", *Transportation Research Part E: Logistics and Transportation Review*, 2015, 81, pp. 75-97, published (Iris et al., 2015b).
- C. Iris, D. Pacino, S. Ropke, A. Larsen, "Integrated Berth Allocation-Quay Crane Assignment Problem: Reformulations, Improved Constraints and Computational Results", *The 4th International Conference on Computational Logistics (ICCL)*, 2013, Copenhagen, presented.
- C. Iris, D. Pacino, S. Ropke, A. Larsen, "Improved Models to Integrated Berth Allocation-Quay Crane Assignment Problem: A Computational Comparison and Novel Solution Approaches", *The 3rd International Conference on Logistics and Maritime Systems (LOGMS)*, 2013, Singapore, presented.

Chapter 3, **Improved formulations and an adaptive large neighborhood search heuristic for the integrated berth allocation and quay crane assignment problem** presents novel valid inequalities and variable fixing methods for the BACAP. Additionally, an Adaptive Large Neighborhood Search (ALNS) heuristic which is based on a destruction-construction cycle is presented for the studied problem. Comparative tests have shown that ALNS can produce high quality solutions both with respect to computational time and solution quality. The construction methods presented in this chapter can be used as a part of other heuristics for such problems. To assess the performance of insertions/removals of vessels and behaviour of the heuristic with different parameters, a number of computational tests are performed. Results are communicated for instances of Meisel and Bierwirth (2009). The work has been disseminated as follows:

- C. Iris, D. Pacino, S. Ropke, "Improved formulations and an adaptive large neighborhood search heuristic for the integrated berth allocation and quay crane assignment problem", 2015, under review (Iris, Pacino, and Ropke, 2015a).
- C. Iris, D. Pacino, S. Ropke, A. Larsen, "Hybrid Heuristic Approaches for Tactical Berth Allocation Problem", *Conference of the International Federation of Operational Research Societies (EURO/IFORS)*, 2014, Barcelona, presented.

Chapter 4, **A two-stage stochastic programming approach to berth and quay crane scheduling problem under uncertainty** focuses on planning seaside operations under uncertainty. Many terminals denote that the information about the exact arrival times and container loads of vessels is sparse. Talks with large and medium scale terminals have shown us that planners decide the berthing, QC assignment and scheduling plans for approximately 3-4 days of planning horizons. Different terminals can handle different numbers of vessels in these intervals, however one thing is clear that the exact arrival time of each vessel and the processing time of operations are uncertain by the time of planning. A stage-wise stochastic programming formulation fits very well for such a problem. We formulate, to the best of our knowledge, the first traditional two-stage stochastic programming model that focuses on berth and QC scheduling problem. We solve the problem with two exact methods one by solving the deterministic equivalent of the stochastic problem and the other by using a variant of Benders decomposition. Computational results reveal that the decomposition approach performs better as the number of scenarios increases and it is competitive with the deterministic equivalent of the stochastic programming model. The novelty also lies in the tests that analyze the value of information for this problem. The stochastic solution helps to obtain important

savings for terminal operators for many instances except very large scale ones where subproblems become computationally intractable. Contributions have also been achieved with the modeling of the problem. Apart from traditional cuts of Benders decomposition, various valid inequalities for the berth and QC assignment problem are presented. This study is conducted during the external research stay at National University of Singapore and container terminals in Singapore have given feedback on the problem definition. The work has been disseminated as follows:

- C. Iris, J.G. Jin, D.H. Lee, S. Ropke, "A two-stage stochastic programming approach to berth and quay crane scheduling problem under uncertainty", 2016, under review, (Iris, Jin, Lee, and Ropke, 2016a).
- C. Iris, J.G. Jin, D.H. Lee, "Quayside Operations Planning Under Uncertainty", *European conference for Operational Research (EURO)*, 2015, Glasgow, presented.

Chapter 5, **A survey on the Ship Loading Problem** reviews the literature on the efficient ship loading process where stowage planning along with loading sequencing and scheduling are integrated to improve the efficiency of the ship loading operations. We formally introduce this problem with the name of Ship Loading Problem (SLP). The study shows that, aside from yard equipment dispatching and scheduling, the number of studies on the optimization of loading operations is limited. Many works have appeared in the past two decades on stowage planning, yet very few focus has been given to the interface of stowage planning with the terminal operations, and those that do, often look at the problem solely from the terminal point of view. This study also emphasizes the need for a public benchmark for the SLP, which could then also be used for its subproblems. The work has been disseminated as follows:

- C. Iris, D. Pacino, "A survey on the Ship Loading Problem", *Computational Logistics, Lecture Notes in Computer Science*, Vol. 9335, pp. 238-251, published (Iris and Pacino, 2015).
- C. Iris, D. Pacino, "A survey on the Ship Loading Problem", *The 6th International Conference on Computational Logistics (ICCL)*, 2015, Delft, presented.

Chapter 6, **Formulations for ship loading problem with transfer vehicle assignment and scheduling** studies the problem introduced in Iris and Pacino (2015) and formulates mathematical models for the problem. The potential

savings of stowage planning with respect to terminal operations have studied by Monaco et al. (2014). In this study, we analyze the integration of transfer vehicle assignment and scheduling to this problem. We formulate a mathematical model to solve the problem and a number of valid inequalities to improve the formulation. Then we suggest a method to obtain the new bounds for the mentioned problem. Computational results show that the enhancements on the model significantly improve the performance of the formulation and the deep integration of these problems significantly helps to obtain cost savings compared to the hierarchical solution methods. The formulations are also converted to solve similar problems with intermediate modifications. A large benchmark is generated for the problem which can be used for various problems related to the SLP. The problem has been studied with a collaborating terminal in Denmark. The methods have also been analyzed on the collaborating terminal data and promising results have been obtained. Due to confidentiality, hypothetical benchmark results will be communicated in the thesis. The work has been disseminated as follows:

- C. Iris, D. Pacino, S. Ropke, "Formulations for ship loading problem with transfer vehicle assignment and scheduling", Technical report, 2016. (Iris, Pacino, and Ropke, 2016b).

The thesis covers the research projects that have been led by Cagatay Iris (the candidate). There are also other supplementary research outcomes during the PhD period. The candidate is actively involved in these supplementary projects but is not the first author of the disseminations. These supplementary research papers are related to the thesis topics, but they are not detailed in the thesis. They are as follows:

- J.G. Jin, C. Iris, "A column generation approach to the robust integrated berth allocation and quay crane assignment problem", Technical report, 2016.
- G. Venturini, C. Iris, A. Larsen, "Multiple-port berth allocation and speed optimization problem: models and computational results", Technical report, 2016.

## 1.6 Conclusion

Mega container terminals have grown to a size where they are called by approximately 60 vessels every day, and they handle more than 30 million TEUs each year. Focusing in Denmark, the largest terminal, that we have, has reached the important edge of 1 million TEUs per year and is called by a mega-container vessel once per week. These volumes are expected to be growing in the future (UNCTAD, 2015). The increasing volumes and complexity of operations have made the job of planners even harder. They alone cannot easily make the best operational plan under these settings, hence they usually rely on some optimization methods to solve these problems.

Traditionally, planning problems have been solved hierarchically depending the information flow between them. However, hierarchical solutions can be poor, misleading or even infeasible for some next stages, so integrating relevant terminal problems has caught the attention of researchers in the field. Integrated container terminal problems, hence, have been receiving more and more attention in the recent years. The work in this thesis contributes to the OR literature on optimization of integrated container terminal problems and has been disseminated in peer-reviewed journals, conferences and technical reports. The contributions cover modeling, methodology, and computational results. There are five research chapters in the thesis, and they are concerned with integrating seaside and yardside operational problems addressing to berth allocation, quay crane assignment, quay crane scheduling, operational stowage planning, transfer vehicle assignment and scheduling problems faced by container terminals.

The integration of terminal problems in this thesis starts from seaside problems and enlarges step by step, in the end it covers the integration of seaside and yardside problems. The first problem is the integrated berth allocation and quay crane assignment problem which determines the berthing position, berthing start/end time for each vessel, and the number of QCs that will be operating on each vessel at each time period during the berthing interval. Planners try to minimize operating costs of QCs and penalty costs due to early/late berthing start/end times. There is a tradeoff between these costs and this tradeoff is investigated with the operational constraints of the problem in Chapter 2 and Chapter 3. The problem definition is also diversified with respect to the properties of the QC assignment and the partitioning of the berthing area. The size of the instances that are solved corresponds to a realistic problem settings. We show that different berth space partitioning methods affect the solution quality, a finer discretization is harder to solve but terminals can better utilize the complete berthing space. Results also present a comparison about the QC assignment policies, and the time-invariant QC assignment is shown to be easier to solve but results in higher costs. Chapter 2 presents novel models that aim



at obtaining an optimal solution to the problem, while Chapter 3 points out improvements on the existing formulations and presents a heuristic method which has acceptable running times for the problem. The integration is deepened in Chapter 4 by incorporating specific QC assignment and scheduling into a variant of the previous problem. The problem studied in this chapter considers the uncertainty in vessel arrival and processing time with the use of scenarios. The problem is decomposed into a master problem which is a berth scheduling problem, and subproblems which are specific QC assignment and scheduling problems of each scenario. The size of instances is diversified for large and small terminal scales and results show that the performance of the solution methods highly depend on the size of the instance. Results also show that solving the problem by considering the uncertainty is extremely important to be able to design a reliable terminal operation.

The next problem focuses on an integrated problem which is composed of sea-side and yardside problems. We integrate problems which are related to the loading operations of the vessel. The loading operation is strongly bounded by the stowage plan which points out the exact position of each container on the vessel. In Chapter 5 and 6, the terminal utilizes the flexibility of making the detailed operative stowage plan with respect to a given class-based stowage plan of the liner shipping company. The flexibility is in selecting the "best" container with respect to terminal logistics for each position by fulfilling the requirement of the class-based stowage plan. Deciding the "best" container with respect to terminal logistics is a problem of load sequencing, transfer vehicle dispatching and scheduling. The objective is to minimize the penalty cost of finishing the loading operations later than an expected time and the cost of operating transfer vehicles during loading operations. Results from a number of formulations show that the integrated problem pays off compared to the hierarchical solution method. Additionally working hours, and consequently, operating costs of transfer vehicles are significantly reduced with the help of integration. It is noted in Minsaas and Psaraftis (2016) that reducing energy consumption of transport vehicles contributes to a more environmentally friendly terminal. Finally such a flexible loading strategy will better help the terminal to finish loading as soon as possible, and this would eventually help to reduce the time that vessel spend at the terminal, consequently transit times for the liner shipping company. As a result, a solution to such a problem will contribute to the Win-Win-Win quest for terminals, liners and the environment in the long run.

## 1.7 Future Work

In the following we outline some possible future research directions related to problems and methodology.

### 1.7.1 Problem Definitions

The integrated problems of Chapter 2 and 3 can be extended to handle additional details such as assignment of specific QCs and their schedules. Park and Kim (2003) show that the assignment of specific QCs is a shortest path problem where decisions about berthing and QC assignment in numbers (BACAP) is pre-processed to the specific QC assignment and scheduling problem. This shortest path problem can be solved by dynamic programming. The deep integration can be studied by integrating this shortest path problem into the BACAP (See properties which show that QC scheduling problem can be solved in polynomial time for special cases in Turkogulari et al. (2016)). Such efforts cover integrating new problems into the BACAP. Instead, variants of the BAP with tidal effects, setup times of QCs, QC properties can be studied in the content of solution methods of Chapter 2 and 3.

Chapter 4 focuses on the planning quayside operations under uncertainty which is already a complicated problem. Rather than focusing on an additional problem, we can incorporate the unexpected QC breakdowns with certain realization probabilities into the scenario tree. A possible breakdown of a QC will not only increase the processing time of a vessel, it will also affect the movement of all QCs around it. The problems studied thusfar are operational problems. A possible future research direction could aim at focusing on tactical berth allocation problem (Giallombardo et al., 2010) under uncertainty. Zhen (2015) focuses on a variant of tactical berth allocation problem under uncertainty where QC assignments are not considered. In this respect, formulations and methods presented in Chapter 4 can be modified to cover tactical planning properties.

Finally we note future research directions regarding the SLP. Chapter 6 studies a variant of the problem where loading sequence and yard handling decisions are fixed for the SLP. The first obvious extension is to optimize the loading sequencing problem within the SLP. This is expected to bring sufficient improvements because large container vessels have many rows in each bay in which QC can load containers with different sequences. Another possible extension is to pool the transfer vehicle for each QC. In the current problem definition, vehicles are dedicated to work on a specific QC throughout the loading period. If the pooling is allowed, efficient routing of vehicles between QCs will reduce the loading

time. Alternatively, effects of dual cycling can be studied on the SLP. The aspects noted thus far are all terminal related. This problem can be further expanded to the decision making problems of shipping liner. Christensen, Pacino, Fonseca, and Psaraftis (2015) focus on the vessels' cargo-mix, in particular finding what cargo composition is needed for a vessel to maximize its utilization on a given service. Depending on the efficiency of loading operations for each container, a feedback loop based integration can be achieved between the two problems.

### 1.7.2 Extensions of Methods

We now briefly note possible ways to extend or improve the methods presented in this thesis. The improvements can be achieved in the solution quality and/or the computational times of the method. Both improvements are useful since better solutions will bring cost reductions and efficient use of resources, while faster methods will allow us to evaluate the decision problems more often. Depending on the problem definition, the planning horizon can be short enough that reducing the computational time of the methods will help to evolve to the real-time decision making.

There are many ways to improve the performance of solution methods of Chapter 2. The first the natural extension is to generate variables dynamically using delayed column generation and solve the set partitioning formulations via a branch-and-price algorithm. Another alternative is analyzing the optimality properties of the set partitioning formulations then reducing and/or ranking the set of columns with these properties (See Rezanova and Ryan (2010) for an example on train driver recovery problem).

Chapter 3 presents various valid inequalities for the BACAP. The first possible extension is to use these inequalities in a Branch-and-Cut algorithm where the relaxed problem is solved. With respect to the ALNS, additional destroy and repair mechanism can be suggested and tested. In the heuristic, the detailed allocation of QC numbers to periods are made by utilizing the Corollary 1 (Iris et al. (2015b)) and a variant of it. This subproblem is an assignment problem which can be solved rapidly by using a set partitioning formulation. This set partitioning problem can be embedded into the repair structure of the ALNS. Once every time the repair method is called, that formulation could be solved. Such a math-heuristic might perform well because the computational results show that weak ALNS solutions are mostly lacking high quality QC number assignments.

The performance of solution method (i.e. the integer L-shaped) in Chapter 4

depends on various factors. The first possible improvement is to use the integer L-shaped method in a Branch-and-Cut fashion (Laporte and Louveaux, 1993) rather than the traditional method in which master problem is solved to optimality in each iteration. Regarding the fact that solving the master problem to optimality at each iteration is computationally very hard, such a method would bring contribution to the results. Recently Boland, Fischetti, Monaci, and Savelsbergh (2015) use the proximity search as a tactical tool to drive Benders decomposition (proximity Benders) and use it a heuristic method to solve two-stage stochastic programming problems, the results are promising. Another future research direction could be applying the proximity Benders for our problem.

Finally we denote future research direction on the solution methods of the SLP. The first potential improvement is to implement a heuristic that utilizes the presented formulations of Chapter 6. The lower bounding model which is very efficient with respect to computational time can be used in a math-heuristic framework. In order to have this framework, it is required to use a neighborhooding structure. Possible neighborhooding methods can be obtained from Kim et al. (2004). Another possible solution extension is to decompose the SLP and generate an intelligent feedback loop between the subproblems.



## CHAPTER 2

# Integrated Berth Allocation and Quay Crane Assignment Problem: set partitioning models and computational results

---

Most of the operational problems in container terminals are strongly interconnected. In this chapter, we study the integrated berth allocation and quay crane assignment problem in seaport container terminals. We will extend the current state-of-the-art by proposing novel set partitioning models. To improve the performance of the set partitioning formulations, a number of variable reduction techniques are proposed. Furthermore, we analyze the effects of different discretization schemes and the impact of using a time-variant/invariant quay crane allocation policy. Computational experiments show that the proposed models significantly improve the benchmark solutions of the current state-of-art optimal approaches.<sup>1</sup>

---

<sup>1</sup>C. Iris, D. Pacino, S. Ropke, A. Larsen, "Integrated Berth Allocation and Quay Crane Assignment Problem: Set partitioning models and computational results", *Transportation Research Part E: Logistics and Transportation Review*, 2015, 81, pp. 75-97, published

## 2.1 Introduction

Fierce global competition in production and trade has forced all entities in supply chains to optimize their logistics operations. The never ending quest for shorter lead times and reduced cost for logistics cost requires extremely efficient logistics systems. In 2013, the worldwide container trade accounts for about 22% of the 6.7 billion tons of dry-cargo trade, and all loads are being transported by vessels via container terminals (UNCTAD (2014)). Recent statistics show that total container trade volumes reached 160 million Twenty-foot Equivalent Units (TEUs) in 2013 with a growth of 4.6% (UNCTAD (2014)). These statistics suggest that logistics efficiency heavily relies on effective container terminal operations. Due to the increasing importance of container terminals and high complexity of their operations, the need for optimization has become evident in recent years. This can also be perceived by the increase of scientific literature where operations research techniques are applied to container terminals (see Stahlbock and Voß (2007) for a survey).

Recent advances in the modeling of container terminal problems have pushed the focus towards integration issues. Conventional hierarchical optimization of sequential operations is known to have possible disadvantages, that may result in infeasible, suboptimal or poor solutions. This is because decisions made in earlier steps were made without considering the resultant knock-on effects in the following stages. This chapter focuses on two important problems on the quayside of terminal operations: the Berth Allocation Problem (BAP) and the Quay Crane Assignment Problem (QCAP). The first problem allocates berthing positions and times for vessels, while the second determines the number of quay cranes (QCs) to be assigned for the load and discharge operations. These two problems are mutually dependent. The number of available QCs depends on where and when the vessel is berthed. Concurrently, the berthing time depends on the processing time of the vessel, which in turn depends on the number of cranes assigned. An integration of those two problems was first introduced by Park and Kim (2003).

The goal of our work is to solve the integrated Berth Allocation and quay Crane Assignment Problem (BACAP), where a berthing time and position for each vessel is assigned during a given planning horizon. A solution to the problem also includes the assignment of QCs, by factoring in marginal productivity losses due to crane interference, and processing times depending on the berthing position of the vessel. An objective is to propose a method that solves instances to optimality. When instances cannot be solved to optimality, tight upper and lower bounds on the objective should be generated. Such bounds can be used to evaluate the performance of future and past heuristics.

An important factor in berth and QC management is the use of policies for the assignment of QCs to vessels. Hence why this chapter analyses two main policies: time-invariant and time-variant QC assignment. The first policy decides how many QCs to assign to a given vessel, and this number cannot change throughout the stay at berth. The second relaxes this assumption and allows the number of assigned QCs to vary during the ship's stay at port. In both cases the number of QCs assigned lies within a given interval specified by the contract between the terminal and the shipping companies. Both variants of the BACAP are modeled in this chapter using a Generalized Set Partitioning (GSPP) formulation. Furthermore, a set of column reduction techniques are presented which help to limit the number of feasible columns generated and to provide better bounds.

The literature on the BAP distinguishes between discrete and continuous versions of the problem with respect to the berth partitioning. In the former version, vessels can only berth at predefined sections of the quay, while this restriction does not apply to the latter version. The work proposed in this study addresses optimal approaches for the continuous case; Where the berth space is discretized in the same manner as Meisel and Bierwirth (2009), Meisel and Bierwirth (2013) and Turkogullari, Taskin, Aras, and Altinel (2014), with berthing at integer points (e.g. every 10 meters). Three discretization techniques are tested in this paper however only one of the them guarantees optimal solutions to the original problem.

In order to validate and evaluate our models, we present a comparison with the BACAP state-of-the-art results by Meisel and Bierwirth (2009). In Meisel and Bierwirth (2009) a compact mathematical model, which can optimally solve some instances of up to 20 vessels, is presented. For larger instances (30, 40 vessels), the model cannot generate integer upper bounds. In the same paper, these upper bounds are generated using different heuristic approaches.

This chapter presents three major contributions: 1) Novel generalized set partitioning formulations that can solve more instances than previous models. 2) Improved upper and lower bounds to almost all instances, bounds that will be of use when testing new algorithms for the problem. 3) Techniques for reducing the number of variables in the model, techniques that can be useful for problems that use a similar modeling approach.

The chapter is organised as follows: First, a literature review is presented in Section 2.2. In Section 2.3, the problem definition and mathematical models proposed by Meisel and Bierwirth (2009) are given. The proposed GSPP models and column reduction techniques are presented in Section 2.4. Extensive computational results are presented and discussed in Section 3.6. The chapter is concluded by Section 2.6.



## 2.2 Literature Review

The importance of container terminal problems has been revealed by many academic studies where authors illustrate recent trends and point out gaps in the literature (see Stahlbock and Voß (2007) for a general review). As regards the integrated quayside problems, recent surveys of Bierwirth and Meisel (2010), Bierwirth and Meisel (2014) focus on berth allocation and QC planning problems (assignment and scheduling) in container terminals. Authors classify berth allocation problems according to spatial, temporal, processing time, and performance indicator attributes. Integration of berth allocation and QC assignment is classified as deep, hierarchical or through a feedback loop. Most papers present compact formulations with deep integration. The BAP remains the main problem and the additional problem is either the assignment or scheduling of QCs. Recently, Meisel and Bierwirth (2013) integrated three of the main seaside terminal planning problems, i.e. BAP, QCAP (in numbers and specific QC assignment) and the quay crane scheduling problem (QCSP).

The BAP is classified as static or dynamic with respect to whether the arrival time of the vessels imposes a bound on the berth start time. One of the first models for dynamic BAP was presented by Imai et al. (2001) and was successively improved by Imai et al. (2005) for the continuous berth allocation case. The latter presents a two-stage heuristic approach which uses discrete berthing solutions and reallocates them in a continuous manner. Cordeau et al. (2005) proposed a Tabu Search (TS) for the dynamic discrete BAP and a continuous variant. A well performing simulated annealing approach is proposed by Kim and Moon (2003) for the continuous BAP.

### 2.2.1 BACAP literature - BAP, QCAP properties

A list of relevant literature for the BACAP is summarized in Table 2.1, in which information about the problem structure, objective function and solution approaches are presented. The studies are listed in chronological order of publication year. In the pioneering paper for the BACAP, Park and Kim (2003) presented a model for the problem. The model supports time-variant QC assignments and is solved by using lagrangian relaxation-based heuristics. Afterwards a dynamic programming method assigns the specific QCs to vessels. With respect to spatial attributes, some papers focus on discrete berth allocation in the BACAP (Imai, Chen, Nishimura, and Papadimitriou (2008), Giallombardo et al. (2010), Vergados, Schaeren, Dullaert, and Raa (2013)). However, continuous berth layout in the BACAP has also attracted many researchers (see Table 2.1). Different extensions appear in the literature surrounding the berth

allocation properties of BACAP. Meisel and Bierwirth (2009) considered the marginal productivity losses due to the QC interference. Experiments with different levels of congestion show the strong impact of the QC-interference on the cost function. The handling time which depends on the berthing position is modeled by Meisel (2009a). Another extension is the modeling of operational constraints of QCs. Giallombardo et al. (2010) proposed a QC profile scheme in which the authors include the effects of shifts, the interference of QCs, the priority of vessels and various real-life constraints. They proposed a two-stage heuristic. In the first stage, QC profiles are assigned to each vessel. In the second stage, authors solve the remaining BAP via a TS heuristic. The BACAP is also studied by Blazewicz, Cheng, Machowiak, and Oguz (2010). They considered the problem as a parallel machine scheduling problem and seek to minimize the makespan.

Problem variations can also be found with respect to the QC assignment. The two main policies are the time-variant and time-invariant QC assignment. In the time-invariant version, authors mostly solve the QC assignment problem first and then solve the BAP (Liang, Huang, and Yang (2009), Chen, Lee, and Cao (2012), etc.). Another modeling aspect is whether individual QCs are assigned or the number of QCs to serve each vessel is determined. Imai et al. (2008) considered the assignment of specific QCs through detailed QC movement constraints. This ensures the assignment of specific QCs, however, the relationship between the number QCs deployed and the processing time could be improved. In another example, Chen et al. (2012) made specific QC-to-vessel assignment and this facilitates the calculation of QC requirements. They proposed valid inequalities that link the berth scheduling and QC assignment better and some valid inequalities are in the form of non-crossing constraints.

### 2.2.2 BACAP literature - Objective function properties

In terms of cost function, we see variations in the modeling of the BACAP. The most popular objective (see Table 2.1) is the composition of berthing costs (QC costs) and time-dependent penalty costs (Chang, Jiang, Yan, and He (2010), Raa, Dullaert, and Schaeeren (2011), Meisel and Bierwirth (2009), etc.). Total weighted service time is another popular objective of the formulations (Liang et al. (2009), Yang, Wang, and Li (2012), etc.). As mentioned in Section 2.2.1, the deviation from expected berthing position may be embedded in the objective with a cost (Chang et al. (2010), Raa et al. (2011), Turkogullari et al. (2014)). Instead of being in the objective, the deviation from expected berthing position might be modeled to affect the processing time (as in Meisel and Bierwirth (2009)). Then the model becomes harder to solve, because the processing time of a vessel would not only depend only on the load of vessel which is mostly

a parameter, it would also depend on a decision variable which is the berthing position.

## 2.2.3 BACAP literature - Solution techniques

The solution approaches are clustered in novel mathematical models, exact methods and heuristic/analytic methods in Table 2.1. Most of the papers propose novel mathematical models for the variants of BACAP. Raa et al. (2011) enrich current models by taking vessel priorities, preferred berthing positions and QC-assignment-dependent handling times into account. The proposed BACAP model is solved using a rolling horizon approach.

### 2.2.3.1 Exact Methods

Several authors use set partitioning formulations to solve different quayside planning problems. The first use of GSPP aimed at solving the BAP (Christensen and Holst (2008)), where the authors proposed a branch-and-price algorithm. The approach can be used to solve both discrete and continuous BAP. For instances of up to 35 vessels, optimal solutions can be found for the discrete BAP, while a gap of 8.3% is obtained for the continuous version. Buhrkal et al. (2011) generated columns a priori and solved the same GSPP model for the discrete case with an IP solver. The approach clearly improved the state-of-the-art and solved the BAP up to 60 vessels to optimality. A recent study by Saadaoui, Umang, and Frejinger (2015) also focuses on the discrete BAP in which 10 berths are considered. They solve a linear programming (LP) relaxation of GSPP model using column generation. When the column generation terminates, they impose the integrality constraints again and resolve the GSPP with the active pool of columns. The framework solves instances of 120 vessels with an average optimality gap of 0.20%. Umang et al. (2013) proposed a GSPP model to solve a more complicated variant of BAP with hybrid berth layout in bulk ports. The model, with a priori generated columns, can solve instances up to 40 vessels to optimality. Robenek et al. (2014) formulated a GSPP model for the integrated berth allocation and yard assignment problem in bulk ports. The problem considers the cargo types on the vessel which affect the storage location in the yard and consequently the berth allocation. They solve the problem with a branch-and-price algorithm. The instances include 10 cargo locations in the yard and 10 berths are available with different equipment. The authors solve instances with 10, 25, and 40 vessels with average optimality gaps of 0.37%, 4.11% and 3.76%, respectively. The branch-and-price is only run for instances of 10 vessels. Due to the time complexity, instances with 25 and 40 vessels are

solved with the column generation and the integrality constraints are imposed in the last stage of column generation to obtain an upper bound.

Vacca et al. (2013) establish the first decomposition method for BACAP and it is based on the model by Giallombardo et al. (2010). The authors suggest a QC profile which holds productivity losses due to QC interferences, vessel priorities and QC assignment for each shift. The authors have implemented a branch-and-price scheme and several accelerating techniques. The approach obtains the optimal results for 10 and 15 vessels and an average gap of 2.95% is obtained for 20 vessels and 5 berths in three hours of time limit.

An exact method to solve the BACAP with continuous (but discretized for each 50 meter) berth layout is presented by Turkogullari et al. (2014). The authors solely consider a time-invariant QC assignment policy. They first formulate a mathematical model to solve BACAP. The model generates optimum solutions up to 60 vessels where there are 24 berthing sections. In addition to that, the authors propose a cutting plane algorithm to solve the BACAP with specific QC-to-vessel assignment by using the optimum solutions of original BACAP model. It is noted that cutting plane algorithm can convert each optimum BACAP solution to the optimum solution of BACAP with specific QC-to-vessel assignment for the instances which are tested.

Chen et al. (2012) have proposed a Benders decomposition method over the berth-level model proposed by Liu et al. (2006). The authors model a reduced version of the BACAP, where the berthing position of each vessel is given, thus leaving the berthing start/end times, specific QC-to-vessel assignment and the positions of QCs as the only decisions to make. The model is decomposed into a master problem and a sub-model, and the results show that the decomposition technique is faster than the original formulation.

In this study, we present exact methods to solve variants of the BACAP (presented in Meisel (2009e) and Meisel and Bierwirth (2009)). Let us now clarify the differences between the BACAP definition used in this paper and that of Vacca et al. (2013) and Turkogullari et al. (2014). The problems considered in Christensen and Holst (2008), Buhkal et al. (2011) and Saadaoui et al. (2015) do not take QC assignment into account, while Chen et al. (2012) assume a partial berth assignment is given. In Vacca et al. (2013), the authors formulate the BACAP with discrete berth allocation, QC moves from one vessel to another are only allowed at the end of the working shifts (restricted time-variant QC assignment). The authors do not consider berthing position dependent processing times, and there are also some differences due to the QC profile definition and the objective function formulation. In Turkogullari et al. (2014), the problem is solely considered for time-invariant QC assignment case of the BACAP. The marginal productivity losses due to the QC interference are not taken into

account. The same is true for the berthing deviation dependent processing times and speeding up option. In their paper, the authors also focus on specific QC-to-vessel assignment problem.

### 2.2.3.2 Heuristic Algorithms

There are also heuristic/analytic approaches which try to solve BACAP. The most popular metaheuristic used to solve BACAP is genetic algorithms (GAs). Imai et al. (2008), Liang et al. (2009), Yang et al. (2012) etc. test various GA configurations which are specific for the defined problem. In the paper by Liang et al. (2009), three different chromosome structures are used to prioritize vessels, to allocate berth, and to assign QC numbers. Chang et al. (2010) formulate the chromosome as a composition of four-dimensional indice pertaining to the arrival sequence, berthing position, berthing time and number of QCs for vessels. Meisel and Bierwirth (2009) propose three heuristics to solve the problem. They conclude that squeaky wheel optimization (SWO) along with local refinements does slightly better than TS. They also show that SWO and TS are better than the First Come First Served-based heuristic. Vergados et al. (2013) propose a Constraint Programming (CP) model with a tailored branching heuristic within a Large Neighborhood Search framework. The model includes QC-to-vessel assignment considering gang (a team of operators that work on QCs) allocations.

The work presented in this chapter builds on the model presented by Meisel and Bierwirth (2009). The literature survey and Table 2.1, show that the model incorporates many relevant constraints and includes several aspects of the problem in its objective function. We therefore believe that the model is a good starting point for the studies carried out in our study.

## 2.3 Modeling the BACAP: Meisel and Bierwirth (2009) Model

Before going into the details of the solution approach, let us introduce the BACAP. We do so by presenting the formulation proposed by Meisel and Bierwirth (2009) and its time-invariant QC assignment version. We propose an extension to this model that allows modeling problems where the number of QCs assigned to a vessel cannot change during the vessel's stay at the berth.

The objective of the BACAP is to find the best berthing position and time for upcoming vessels by fulfilling the QC requirement of each vessel. A solution for

Table 2.1: Integrated BACAP literature abstract

		Problem Structure										Objective Function										Solution Approach													
Year	Authors	A		B	C	D	E	F	G	Type		$T_w$	E	T	L	$\Delta b$	$C_{qc}$	$C_k$	X	Y			Z												
2003	Park and Kim (2003)	1	2	3	1	2	Y	1	2	1	2	Min	*		*	*	*	*	1	1	2	3	1	2	3	4	5	6	7						
2008	Imai et al. (2008)	*			*							Min	*			*			*				*												
2009	Liang et al. (2009)		*	*								Min											*												
2009	Meisel and Bierwirth (2009)		*	*		*						Min	*	*	*	*	*	*	*					*	*	*	*	*	*						
2010	Giallombardo et al. (2010)		*	*	*	*	*	*	*	*	Max		*			*	*	*	*					*	*	*	*	*	*						
2010	Chang et al. (2010)		*	*	*	*	*	*	*	*	Min		*			*	*	*	*					*	*	*	*	*	*						
2011	Raa et al. (2011)										Min					*	*	*	*					*	*	*	*	*	*						
2011	Blazewicz et al. (2010)	*			*	*	*	*	*	*	Min	*				*	*	*	*					*	*	*	*	*	*						
2012	Yang et al. (2012)	*	*	*	*	*	*	*	*	*	Min	*				*	*	*	*				*	*	*	*	*	*	*						
2012	Chen et al. (2012)		*	*	*	*	*	*	*	*	Min		*	*	*	*	*	*	*				*	*	*	*	*	*	*						
2013	Vacca et al. (2013)	*	*	*	*	*	*	*	*	*	Max					*	*	*	*				*	*	*	*	*	*	*						
2013	Vergados et al. (2013)	*	*	*	*	*	*	*	*	*	Min				*	*	*	*	*				*	*	*	*	*	*	*						
2014	Turkogullari et al. (2014)	*	*	*	*	*	*	*	*	*	Min		*	*	*	*	*	*	*				*	*	*	*	*	*	*						

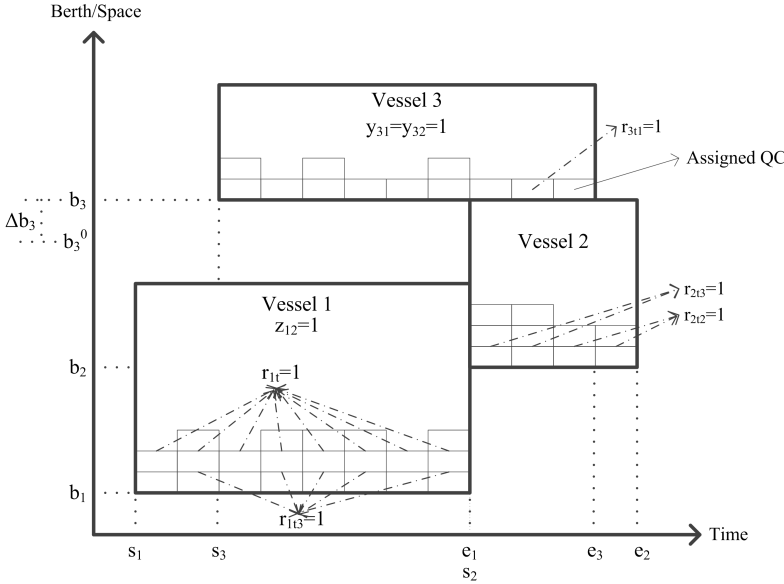
Problem Structure = {A: Temporal attribute: (1: Static, 2: Dynamic, 3: Rolling horizon), B: Spatial attribute: (1: Discrete, 2: Continuous), C: Interference, D: Deviation depending handling times, E: Operational constraints: Columns with "Y" heading means "Yes, the attribute is taken into account".

F: QC assignment: (1: Time-variant, 2: Time-invariant), G: QC policy: (1: QC-to-vessel, 2: The number of QCs to assign)}  
 Objective Function =  $\{T_w$ : Total weighted service time (handling, waiting, etc.),  $E$ : Earliness,  $T$ : Tardiness,  $L$ : Lateness,  $\Delta b$ : Deviation from expected berthing position,  $C_{qc}$ : Cost of QC assignment or changing QC-plan,  $*$  represents gains earned through QC plans,  $C_k$ : Cost of housekeeping or other operations}

Solution Approaches = {X:1: Novel mathematical models, Y: Exact Methods: (1: Lagrange relaxation, 2: Branch-and-Price, 3: Other exact methods), Z: Heuristic, Simulation, AI approaches: (1: Genetic algorithms, 2: Squeaky wheel optimization, 3: Tabu search, 4: Greedy/LNS heuristics, 5: Simulation, 6: Approximation heuristics, 7: Constraint programming)}

each vessel determines the berthing position, the berthing start, end times and the number of QCs that are operating on the vessel at any given time. The time horizon is discretized. Any discretization can be used but it is useful to think of a discretization into whole hours. The berthing position is determined by a continuous variable, but the data used with the model ensures that ships are berthed at integer positions.

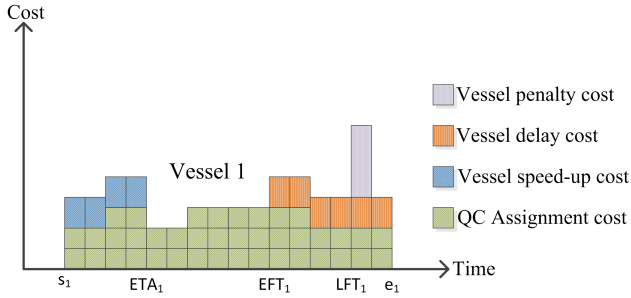
An example of a BACAP plan can be seen in Figure 2.1 that shows the berthing plan in a time/space diagram. In this example, three ships are berthed in the harbor. Each vessel is represented by a rectangle that shows the time and space occupied by the vessel. The smaller rectangles indicate the assignment of QCs to vessels, each small rectangle represents one QC. Each ship has an upper and lower limit on the number of cranes that can be assigned to it. These bounds are determined by contracts between the vessel owner and the port and by the size of the ship. A limited number of QCs are available in the harbor and this determines the maximum number of cranes that we can assign in any time slot. The symbols used on the Figure 2.1 will be explained in the following section.



**Figure 2.1:** Berth-time diagram of BACAP.

The objective function is a combination of time-dependent costs and QC assignment costs. The time dependent costs can be attributed to the berthing start and end times while the QC dependent costs are a function of how many QCs

are assigned to each vessel. If a vessel is berthed before its Expected Time of Arrival (ETA), a speed-up cost occurs for each rushed time unit. Delay costs depend on how many time units have passed from the Expected Finishing Time (EFT). An ulterior penalty cost incurs if the berthing end time is beyond the Latest Finishing Time (LFT). An example of the cost structure of vessel 1 can be seen in Figure 2.2. The figure shows that during the vessel's stay, between its start ( $s_1$ ) and end time ( $e_1$ ), QC operations costs are distributed along periods depending on the number of QCs operating. Since the vessel start time is four periods earlier than its ETA ( $ETA_1$ ), speed-up costs occur. In the same manner, we must pay delay costs due to the vessel's end time being later than the EFT ( $EFT_1$ ). Finally, since operations go beyond the LFT ( $LFT_1$ ), a one-time penalty is added to the cost.



**Figure 2.2:** Cost structure of BACAP for Vessel 1 in Figure 2.1.

### 2.3.1 Time-variant model

Let us now introduce the model proposed by Meisel and Bierwirth (2009). Table 6.1 presents the mathematical notation.



**Table 2.2:** BACAP mathematical notation

Parameters and sets:	
$V$	Set of all vessels to be served, $V \in \{1, 2, \dots, N\}$ , where $N$ is the number of vessels to be planned
$L$	Length of the quay
$T$	Set of 1 hour periods, $T \in \{0, 1, \dots, H - 1\}$ , where $H$ is the end of planning horizon
$l_i$	Length of vessel $i \in V$
$b_i^0$	Desired berthing position of vessel $i \in V$
$m_i$	Quay crane capacity demand of $i \in V$ given as QC-hours
$r_i^{min}$	Minimum number of QCs agreed to serve vessel $i \in V$ simultaneously
$r_i^{max}$	Maximum number of QCs agreed to serve vessel $i \in V$ simultaneously
$R_i$	Feasible range of QCs assignable to vessel $i \in V$ , where $R_i = [r_i^{min}, r_i^{max}]$
$ETA_i$	Expected arrival time of vessel $i \in V$
$EST_i$	Earliest time of arrival of vessel $i \in V$ when it is speed up
$EFT_i$	Expected finishing time of vessel $i \in V$
$LFT_i$	Latest finishing time of vessel $i \in V$ without any penalty cost
$c_i^1$	Speed up cost of vessel $i \in V$ on its journey to catch a berthing time earlier than $ETA_i$
$c_i^2$	Cost of exceeding the expected finishing time $EFT_i$ for vessel $i \in V$
$c_i^3$	Penalty cost by exceeding $LFT_i$ for vessel $i \in V$
$c^4$	Cost rate per QC-hour of operations
$\alpha$	Interference exponent for the QCs. Only $q^\alpha$ effective QC hours are obtained when assigning $q$ QCs to a ship for one hour
$\beta$	Berth deviation factor. A ship $i$ placed at position $b_i$ needs $(1 +  b_i^0 - b_i \beta)m_i$ effective QC hours. $ b_i^0 - b_i $ is the deviation from desired berthing position
$M$	A large positive number
$Q$	Available number of QCs
Decision variables:	
$b_i \in \mathbb{Z}^+$	Berthing position of vessel $i \in V$
$s_i \in \mathbb{Z}^+$	Time of starting the handling (berthing start time) of vessel $i \in V$
$e_i \in \mathbb{Z}^+$	Time of ending the handling (berthing end time) of vessel $i \in V$
$r_{it} \in \mathbb{B}$	1; if there is any QC assignment to vessel $i$ in period $t$ , 0 otherwise
$r_{itq} \in \mathbb{B}$	1; if there is exactly $q$ QC assigned to vessel $i$ in period $t$ , 0 otherwise
$\Delta b_i \in \mathbb{Z}^+$	Deviation from desired berth if vessel $i$ is in position $b$ , $\Delta b_i =  b_i^0 - b_i $
$\Delta ETA_i \in \mathbb{Z}^+$	Required speedup to reach start-time $s_i$ by vessel $i$ , where $\Delta ETA_i =  ETA_i - s_i $
$\Delta EFT_i \in \mathbb{Z}^+$	Tardiness of vessel $i \in V$ when operations are finished later than expected finishing time, $\Delta EFT_i =  e_i - EFT_i $
$u_i \in \mathbb{B}$	1; if finishing time of vessel $i \in V$ exceed latest finishing time, 0 otherwise
$y_{ij} \in \mathbb{B}$	1; if vessel $i$ is berthed below vessel $j$ in berth area, i.e. $b_i + l_i \leq b_j$ , 0 otherwise
$z_{ij} \in \mathbb{B}$	1; if handling of vessel $i$ ends no later than handling of vessel $j$ starts in berth area, 0 otherwise
Time invariant decision variables:	
$p_{iq} \in \mathbb{B}$	1; if $q$ QCs are assigned to vessel $i$ , 0 otherwise

$$\min \sum_{i \in V} (c_i^1 \Delta ETA_i + c_i^2 \Delta EFT_i + c_i^3 u_i + c^4 \sum_{t \in T} \sum_{q \in R_i} r_{itq} q) \quad (2.1)$$

subject to

$$\sum_{t \in T} \sum_{q \in R_i} q^\alpha r_{itq} \geq (1 + \Delta b_i \beta) m_i \quad \forall i \in V \quad (2.2)$$

$$\sum_{i \in V} \sum_{q \in R_i} q r_{itq} \leq Q \quad \forall t \in T \quad (2.3)$$

$$\sum_{q \in R_i} r_{itq} = r_{it} \quad \forall i \in V, \forall t \in T \quad (2.4)$$

$$\sum_{t \in T} r_{it} = e_i - s_i \quad \forall i \in V \quad (2.5)$$

$$(t + 1) r_{it} \leq e_i \quad \forall i \in V, \forall t \in T \quad (2.6)$$

$$r_{it} t + H(1 - r_{it}) \geq s_i \quad \forall i \in V, \forall t \in T \quad (2.7)$$

$$\Delta b_i \geq b_i - b_i^0 \quad \forall i \in V \quad (2.8)$$

$$\Delta b_i \geq b_i^0 - b_i \quad \forall i \in V \quad (2.9)$$

$$\Delta ETA_i \geq ETA_i - s_i \quad \forall i \in V \quad (2.10)$$

$$\Delta EFT_i \geq e_i - EFT_i \quad \forall i \in V \quad (2.11)$$

$$Mu_i \geq e_i - LFT_i \quad \forall i \in V \quad (2.12)$$

$$b_j + M(1 - y_{ij}) \geq b_i + l_i \quad \forall i, j \in V, \quad i \neq j \quad (2.13)$$

$$s_j + M(1 - z_{ij}) \geq e_i \quad \forall i, j \in V, \quad i \neq j \quad (2.14)$$

$$y_{ij} + y_{ji} + z_{ij} + z_{ji} \geq 1 \quad \forall i, j \in V, \quad i \neq j \quad (2.15)$$

$$s_i, e_i \in \{EST_i, \dots, H\} \quad \forall i \in V \quad (2.16)$$

$$b_i \in \{0, 1, \dots, L - l_i\} \quad \forall i \in V \quad (2.17)$$

$$\Delta ETA_i, \Delta EFT_i \geq 0 \quad \forall i \in V \quad (2.18)$$

$$r_{itq}, r_{it}, u_i, y_{ij}, z_{ij} \in \{0, 1\} \quad \forall i, j \in V, \forall t \in T, \forall q \in R_i \quad i \neq j \quad (2.19)$$

The objective function (6.1) is a minimization of the overall cost which has two major components. The first, is based on the vessels' time at port (speed-up cost, delay cost, and penalty cost). The second, is linked to the QC assignments in which the number of QCs used is multiplied by the cost rate per QC-hour. Constraint (3.2) ensures that every vessel receives the required QC capacity, taking into account productivity losses by QC interference, and increased QC demand due to deviation from the expected berthing position. Constraint (3.3) enforce restrictions so that the assigned QC number cannot exceed the available number of QCs in the terminal. Constraint (3.4) links the  $r_{itq}$  and  $r_{it}$  variables: if any  $q$  QCs are assigned to vessel  $i$  in period  $t$ , then operations are ongoing

on the vessel which should therefore stay at berth. Constraints (3.5), (3.6) and (3.7) link the  $r_{it}$  variables with the arrival and departure variables. The constraints guarantee that the  $r_{it}$  variables are only set to one when  $t \in [s_i, e_i]$  and that operations are not preemptive. Constraint (3.8)-(3.12) determine the deviation from the expected berthing place, the required speed-up for vessel to reach  $s_i$ , the tardiness of the operations, and sets  $u_i$  to one if the ship departs after  $LFT_i$ . Constraint (3.13) and (3.14) are used to set the variables  $y_{ij}$  and  $z_{ij}$ . These variables are used in constraint (3.15) to avoid that ships overlap in time or space. Definition of domains (3.16) and (3.17) ensure that start-time and end-time of operations are between the  $ETA_i$  and the end of the planning horizon. The berthing position of vessel  $i$  is restricted by the berth and the vessel length. Constraints (3.18) and (4.7) define the domains of the remaining variables.

### 2.3.2 Time-invariant model

In the model presented in Section 2.3.1 the number of QCs assigned to a vessel can change over time. However, some terminals may opt not to change the number of QCs throughout the vessel's stay at port, in order not to create additional congestion of QC rescheduling, and therefore we propose a variant of the model where the number of QCs assigned to a vessel is fixed throughout the vessel's stay. Mind that the number of cranes to assign is still a decision variable. This problem has been studied in academic literature and is named time-invariant BACAP (Turkogullari et al. (2014), Yang et al. (2012), Meisel (2009e), etc.). To model the time-invariant QC assignment we add the binary decision variable  $p_{iq}$  which is one iff  $q$  QCs are assigned to vessel  $i$ . The time-invariant QC assignment is then enforced by the following constraints:

$$\sum_{q \in R_i} p_{iq} = 1 \quad \forall i \in V \quad (2.20)$$

$$r_{itq} \leq p_{iq} \quad \forall i \in V, \forall t \in T, \forall q \in R_i \quad (2.21)$$

$$p_{iq} \in \{0, 1\} \quad \forall i \in V, \forall q \in R_i \quad (2.22)$$

Constraint (2.20) ensures that exactly one QC number is chosen for each vessel  $i$ . Constraint (2.21) links the  $r_{itq}$  and  $p_{iq}$  variables. If  $q$  QCs are assigned to vessel  $i$  ( $p_{iq} = 1$ ),  $r_{itq}$  is either one or zero. We have to allow  $r_{itq} = 0$  since there are some periods  $t$  where the vessel is not at berth. If  $p_{iq}$  equals zero the corresponding  $r_{itq}$  are forced to zero through the entire planning horizon. Constraint (3.5) guarantees avoiding preemption by preventing any zero values

for  $r_{itq}$  within the berthing interval. Constraint (2.20), along with Constraints (3.5-3.7), ensure that only a fixed number of QCs is used without preemption in operations.

## 2.4 Generalized Set Partitioning Formulations

In this section, we present GSPP reformulations for the time-variant and time-invariant BACAP. These models are based on models for the berth allocation problem presented by Christensen and Holst (2008) (see also Buhrkal et al. (2011)). The addition of QC decisions is, to the best of our knowledge, novel. The proposed models contain a large number of variables and it is tempting to use column generation to solve them. However, in this paper we use the simpler approach of generating all variables a priori (as it also was successfully done in Buhrkal et al. (2011)).

### 2.4.1 Time-Invariant GSPP Model

In the time-invariant GSPP model, a column represents a feasible assignment of a single vessel to a position in time and space (recall Figure 2.1), as well as an assignment of QCs for the berthing period. In addition to the already introduced notation, we introduce some additional notation. The set of columns (assignments) is denoted by  $\Omega$ . We define three matrices  $(a_{ij})$ ,  $(b_{pj})$  and  $(q_{tj})$ , all containing  $|\Omega|$  columns. Matrix  $(a_{ij})$  contains a row for each vessel. Each element  $a_{ij}$  is binary and it is 1 iff column  $j$  represents an assignment of vessel  $i \in V$ . Each element of  $(a_{ij})$  contains exactly one non-zero element. Binary matrix  $(b_{pj})$  contains a row per (berth, time) position. The entry  $b_{pj}$  is one iff position  $p \in P$  is occupied in the assignment that variable  $y_j$  represents. The matrix  $(q_{tj})$  contains a row per time unit. An element  $q_{tj}$  indicates the number of QCs that are assigned to vessel  $j$  in time period  $t$ . Since we are modeling the time-invariant version of the problem, each column contains zeroes and one or more copies of a number  $\tilde{q}$  which indicate the number of QCs used in the assignment that the variable represent. Each column  $j$  has a cost  $c_j$ . This cost is easily calculated from the vessel index, the (berth, time) position and the QC allocation. In the GSPP model, the berth dimension is discretized into  $S$  berth cells. Each vessel can occupy multiple cells when the discretization is fine enough. We let  $P$  be the set of (berth, time) positions that a ship can occupy. This set contains  $H \cdot S$  elements. The decision variables of the models are denoted  $y_j, j \in \Omega$ , they are binary and indicate whether column (assignment)  $j$

should be used in the solution. The model is:

$$\min \sum_{j \in \Omega} c_j y_j \quad (2.23)$$

subject to

$$\sum_{j \in \Omega} a_{ij} y_j = 1 \quad \forall i \in V \quad (2.24)$$

$$\sum_{j \in \Omega} b_{pj} y_j \leq 1 \quad \forall p \in P \quad (2.25)$$

$$\sum_{j \in \Omega} q_{tj} y_j \leq Q \quad \forall t \in T \quad (2.26)$$

$$y_j \in \{0, 1\} \quad \forall j \in \Omega \quad (2.27)$$

The objective function (2.23) minimizes the sum of the costs for the selected variables. Constraint (2.24) guarantees that all vessels are served. Constraint (2.25) restricts each berth/time position to be used at most once. Constraint (2.26) ensures that we do not use more QCs than are available at the container terminal.

We illustrate the model with a small example containing two vessels, the first with a length of one and the second with a length of two berth units. Vessels 1 and 2 have a requirement of 2 and 4 QC hours, respectively. In this example we disregard that interference and a bad positioning can increase QC capacity demand. Furthermore, Vessel 1 has  $\{r_i^{min}, r_i^{max}\} = \{1, 2\}$ , and vessel 2 has  $\{r_i^{min}, r_i^{max}\} = \{3, 5\}$ . The earliest berthing start times ( $EST_i$ ) for the two vessels are 1 and 2. Additionally, we assume that there are two berthing spaces, three planning periods, and six QCs available to serve the vessels. For the first vessel, all feasible solutions are presented in Table 2.3, while for the second, only a small portion is illustrated. The first two rows indicate which vessel the column is representing. The next six rows represent the 6 available time/space positions and indicate which position each assignment occupies. The last three rows indicate how many QCs are used in each time period by the assignment. The 15 columns with heading  $y_j$  indicate 15 possible assignments for vessel 1 and 2 while the column RHS gives the right hand side of each constraint. The last column simply indicates the mathematical representation (symbol) of the columns in the model. Note that some of the columns presented in Table 2.3 might be removed by the column reduction techniques which will be discussed later.

The simple structure of the model is convenient, but its drawback is that the

**Table 2.3:** Structure of assignment matrix for given example

$j =$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	RHS	
Vessel 1	1	1	1	1	1	1	1	1	1	1						$= 1$	$a_{ij}$
Vessel 2											1	1	1	1	1	$= 1$	
Space1/Time1	1				1											$\leq 1$	$b_{pj}$
Space1/Time2	1	1				1					1	1		1		$\leq 1$	
Space1/Time3		1					1				1		1		1	$\leq 1$	
Space2/Time1			1					1								$\leq 1$	
Space2/Time2			1	1					1		1	1		1		$\leq 1$	
Space2/Time3				1						1	1		1		1	$\leq 1$	
Time 1	1		1		2			2								$\leq 6$	$q_{tj}$
Time 2	1	1	1	1		2			2		3	4		5		$\leq 6$	
Time 3		1		1			2			2	3		4		5	$\leq 6$	

model can contain many variables (dependent on choice of planning horizon and discretization of the berth space). This model also handles the case where the number of QCs assigned to a vessel vary from time-period to time-period. This variant can be represented by allowing the entries in each column of the  $(q_{tj})$  matrix to take more than two values.

Modeling the time-variant QC assignment this way, however, will increase the number of variables dramatically, therefore we have not pursued this direction. Instead, a different modeling approach for the time-variant number of QCs is presented in Section 2.4.2.

### 2.4.2 Time-variant GSPP Model

As for the time-invariant model, a column for the time-variant GSPP formulation represents a feasible assignment of a single vessel to a berth with its expected processing time. The difference, compared to the model from Section 2.4.1, is on how the processing times and QC assignment are handled. The exact number of QCs to serve the vessel in each period is not embedded in the column representation. Alternatively, since we know the minimum and maximum number of QCs that can serve a vessel in parallel  $(r_i^{min}, r_i^{max})$ , we can calculate the minimum and maximum processing time for a given ship at a given position (recall that position impacts processing time through the  $\beta$  parameter). Then, we proceed to generate an assignment for each possible processing time. The set of columns is again denoted by  $\Omega$ . We define two matrices  $(a_{ij})$ ,  $(b_{pj})$  which contain  $|\Omega|$  columns.  $a_{ij}$  and  $b_{pj}$  are interpreted in the same way as in Section 2.4.1.  $\Omega^B(b, i)$  is the set of columns that places the start of ship  $i$  in berth  $b$  (so a column will only occur in one of the sets  $\Omega^B(b, i)$  even if it takes up several berths).  $\Omega^T(t, i)$  is the set of columns (assignments) that represent a placement

of ship  $i$  that occupies time period  $t$ . Each column  $j$  has a cost value  $c_j$ . This cost includes the cost components which are related to the timing of the vessel (too early/too late), but leaves out the component related to the number of QCs used (see (6.1)), since this information cannot be deduced from the information in the column. There are two sets of decision variables:  $y_j$  determines if the column  $j$  is used or not, while  $r_{itq}$  is a binary variable that is 1 if  $q$  cranes are assigned to vessel  $i$  at time  $t$ . The additional parameters and notations that are not listed in Section 2.3 and 2.4.1 are as follows:

Additional set notations for GSPP model:	
$P$ :	Set of positions: a position is a pair (berth, time slot), $P \in \{0, 1, \dots, H \cdot S\}$
$B$ :	Set of berthing spaces, $B \in \{1, 2, \dots, S\}$
$\Omega^B(b, i)$ :	The set of columns representing a placement of ship $i$ in berth $b$ .
$\Omega^T(t, i)$ :	The set of columns representing a placement of ship $i$ that occupies time period $t$ .
Additional parameters:	
$\underline{t}_{i,b}$ :	Minimum time (number of time periods needed to serve ship $i$ in berth $b$ ):
$\underline{t}_{i,b} = \left\lceil \frac{(1+\beta\Delta_{bi})m_i}{(r_i^{max})^\alpha} \right\rceil$	
$\bar{t}_{i,b}$ :	Maximum time (number of time periods needed to serve ship $i$ in berth $b$ ):
$\bar{t}_{i,b} = \left\lceil \frac{(1+\beta\Delta_{bi})m_i}{(r_i^{min})^\alpha} \right\rceil$	
$\Delta_{bi}$ :	The absolute distance of berthing place $b$ from desired position of vessel $i$ :
$\Delta_{bi} =  b_i^0 - b $	
Decision variables:	
$y_j \in \{0, 1\}$ :	1 if column $j$ (a certain ship/position/duration combination) is used, 0 otherwise
$r_{itq} \in \{0, 1\}$ :	1 if $q$ cranes are assigned to ship $i$ at time $t$ , 0 otherwise

Hence, the mathematical model can be formulated as:

$$\min \sum_{j \in \Omega} c_j y_j + c_4 \left( \sum_{\forall i \in V} \sum_{\forall t \in T} \sum_{q \in R_i} q r_{itq} \right) \quad (2.28)$$

subject to

$$\sum_{j \in \Omega} a_{ij} y_j = 1 \quad \forall i \in V \quad (2.29)$$

$$\sum_{j \in \Omega} b_{pj} y_j \leq 1 \quad \forall p \in P \quad (2.30)$$

$$\sum_{i \in V} \sum_{q \in R_i} q r_{itq} \leq Q \quad \forall t \in T \quad (2.31)$$

$$\sum_{t \in T} \sum_{q \in R_i} q^\alpha r_{itq} \geq \sum_{b \in B} \left( (1 + \Delta_{bi} \beta) \sum_{j \in \Omega^B(b, i)} y_j \right) \quad \forall i \in V \quad (2.32)$$

$$\sum_{q \in R_i} r_{itq} = \sum_{j \in \Omega^T(t, i)} y_j \quad \forall i \in V, t \in T \quad (2.33)$$

$$y_j \in \{0, 1\} \quad \forall j \in \Omega \quad (2.34)$$

$$r_{itq} \in \{0, 1\} \quad \forall i \in V, t \in T, q \in R_i \quad (2.35)$$

The objective (2.28) is formulated as a sum of column costs (which include speedup, lateness and penalty costs) and QC assignments costs. Constraint (2.29) and (2.30) are the same as (2.23)-(2.27). Constraint (2.31) guarantees that at most  $Q$  QCs are used in each time period. The next two constraints (2.32) and (2.33) link the columns used and the QC assignment variables in terms of berthing time and position. Constraint (2.32) ensures that enough QC capacity is assigned to each vessel. The left hand side of constraint (2.32) measures the number of effective QC hours assigned to the vessel and takes the interference factor into account. The right hand side calculates how many effective QC hours are necessary and takes the berthing position into account. Constraint (2.33) imposes that QCs are only assigned to a vessel when it is at port. This constraint also imposes that at most one QC assignment policy can be applied ( $\sum_{q \in \bar{Q}} r_{itq} \leq 1$  where  $\bar{Q} \in \{1, 2, \dots, Q\}$ ) for each vessel in each period. What is more, constraint (2.33) guarantees that the QC assignment is non-preemptive in the periods between the column start and end interval.

In the time-variant model, not only the number of constraints, but also the number of columns is higher compared to the time-invariant model. In the time-invariant GSPP, we typically generate  $r_i^{max} - r_i^{min}$  columns for each ship and position (berth/time) combination. For the time-variant case it is  $\bar{t}_{i,b} - \underline{t}_{i,b}$ . Since the QC requirements ( $m_i$ ) are rather high in the instances, we typically have  $\bar{t}_{i,b} - \underline{t}_{i,b} > r_i^{max} - r_i^{min}$  and the time-variant GSPP thus needs a larger number of columns.

The GSPP models offer some modeling advantages compared to compact models like the one presented in Section 2.3.1. GSPP models allow the handling



of many types of constraints implicitly while generating the feasible columns. Also various objective functions can easily be handled as long as they can be decomposed into a cost per column.

### 2.4.3 Discretization policies

The complexity of GSPP and the number of columns vary by using different discretization policies of the continuous berth space. In the original data set proposed by Meisel and Bierwirth (2009), the length of the berth is 100 units (1000m: 100x10m segments). In this study, three approaches are proposed to test the performance of the formulations:

- Berth Length of 1: In this representation, we have 100 berthing spaces ( $S$ ) where each of them has a 1 unit of length ( $l_s$  : 10m in real-life). This formulation directly corresponds to the version studied by Meisel and Bierwirth (2009).
- Berth Length of 2: In this version, we have 50 berthing spaces ( $S$ ) where each of them has 2 units of length ( $l_s$  : 20m in real-life). This representation results in a smaller model but the solution quality is decreased since we cannot use the quay space as efficiently as in the BL=1 approach. This discretization is inspired by the distance between bollards at the port.
- Dynamic (Hybrid) Discretizing: In this approach, we let the discretization length be dependent on the specific vessel. The policy is derived from the observation that the berthing of the optimal solution for vessel  $i$  usually lies around its desired berthing position ( $b_i^0$ ). Hence, we do a finer discretization for 5 berthing spaces around the desired berthing position, for the rest of the berth length, a discretization of 2 is used. This policy usually results in around 55 berthing spaces depending on whether  $b_i^0$  is close to the start or the end of the berth.

The different discretization policies are tested in Section 2.5.3 for both the time-variant and time-invariant case.

### 2.4.4 Column Reduction Strategies and Valid Inequalities for set partitioning models

The number of necessary columns may be very large when dealing with a high number of vessels and a fine discretization. Hence, in this section we propose

some rules for eliminating columns that cannot be part of the optimal solution. This decreases memory consumption and makes the model easier to solve.

In each column reduction technique, an upper bound (UB)  $\bar{z}$  is required for the value of the objective function (6.1). By having this bound, we can decide whether to keep a column or simply remove it. The upper bound can be obtained using a heuristic for the BACAP, for now it is simply assumed that an upper bound is known.

#### 2.4.4.1 Preprocessing-1: Simple redundancy

Given the upper bound  $\bar{z}$  on the objective value, a simple but nevertheless useful preprocessing rule is to remove columns with cost  $c_j > \bar{z}$ . This applies to both GSPP models. But for the time-variant version, a better bound can be obtained since  $c_j$  does not include the QC component. To do so, we calculate a lower bound (LB) on the costs of the QC assignments. First, the minimum number of crane hours needed ( $\theta$ ) is calculated by (2.36).

$$\theta = \sum_{i \in V} r_i^{\min} \left\lceil \frac{m_i}{(r_i^{\min})^\alpha} \right\rceil \quad (2.36)$$

Given a vessel, the shortest possible processing time (when  $r_i^{\min}$  vessels are assigned) can be calculated with  $\left\lceil \frac{m_i}{(r_i^{\min})^\alpha} \right\rceil$ , we then multiply this with the ship's minimum number of required QCs. This is obviously a lower bound on the number of QC hours needed for the ship and it is easy to calculate. Using  $\theta$ , we calculate a lower bound on the QC component of the objective using  $\underline{z} = c^4 \theta$  and all columns with  $c_j + \underline{z} > \bar{z}$  can be removed.

#### 2.4.4.2 Preprocessing-2: Contribution regarding Lower Bound (LB)

The second preprocessing procedure is based on calculating a lower bound on the total objective by selecting the "best" column for each ship. For each ship we calculate the increased lower bound caused by selecting a column  $j$  instead of the vessel's best column. If that lower bound is greater than the upper bound, column  $j$  can be discarded. In the following, the idea is explained in more detail. We first describe the procedure for the time-invariant case, since this is the simplest.

Let  $\Omega(i)$  be the columns corresponding to vessel  $i$ , then we can calculate  $\sigma_i$ , the lowest column cost for columns representing ship  $i$  by:

$$\sigma_i = \min_{j \in \Omega(i)} \{c_j\} \quad (2.37)$$

A lower bound on the overall objective is then:

$$\underline{z}^2 = \sum_{i \in V} \sigma_i \quad (2.38)$$

Let  $\tau(j)$  be the vessel associated with column  $j$ , then any column  $j$  for which  $\underline{z}^2 + c_j - \sigma_{\tau(j)} > \bar{z}$  can be removed. The left hand side (LHS) computes the lower bound on the objective if column  $j$  is used instead of the best column for ship  $\tau(j)$ .

The preprocessing rule also works for the time-variant GSPP, but in this case it can be improved since  $c_j$  does not contain the QC component. Let  $\epsilon(i, d, \Delta b)$  be a lower bound on the number of QC hours needed to serve ship  $i$  when berthed  $\Delta b$  units away from the desired position and having a stay of  $d$  time units at port. With this we can calculate an improved lower bound  $\phi(j)$  for column  $j$ 's contribution to the objective function:

$$\phi(j) = c_j + c_4 \epsilon(\tau(j), d(j), \Delta b(j)) \quad (2.39)$$

where  $d(j)$  and  $\Delta b(j)$  are the duration of the port stay and the deviation from best berth position for column  $j$ , respectively. We now use  $\phi(j)$  to define the lowest contribution  $\sigma_i$  for each ship  $i$ :

$$\sigma_i = \min_{j \in \Omega(i)} \{\phi(j)\} \quad (2.40)$$

and we compute the lower bound on the total objective as before:  $\underline{z}^2 = \sum_{i \in V} \sigma_i$ . A column can now be eliminated if  $\underline{z}^2 + \phi(j) - \sigma_{\tau(j)} > \bar{z}$ .

What remains, is to describe how we calculate the lower bound  $\epsilon(i, d, \Delta b)$ . When a vessel is placed  $\Delta b$  positions away from the desired position, we have to put in  $(1 + \Delta b \beta)m_i$  raw crane hours to serve the vessel. To minimize the number of QC hours needed, we have to spread the work evenly during vessel's stay interval to avoid high interference factors. We would have to work for  $d^1$  hours with  $x$  QCs and for  $d^2$  hours with  $x + 1$  QCs. A method to calculate the  $\epsilon$  function is presented in Algorithm 5.

The idea behind the procedure is identifying available capacity gaps in given processing times. By knowing the processing time of a given vessel  $i$ , we can calculate how many periods corresponds to which number of QCs in a solution.

**Algorithm 1:** Approximation of  $\epsilon$ 


---

```

1 Require:  $i, r_i^{min}, r_i^{max}, d(j), (1 + \beta\Delta b(j))m_i$ 
2   if  $(d(j)(r_i^{min})^\alpha \geq (1 + \beta\Delta b(j))m_i)$ 
3     return  $d(j)r_i^{min}$ ;
4   else
5     Find  $q \in \{r_i^{min}, \dots, r_i^{max}\}$  such that
       $d(j)q^\alpha \leq (1 + \beta\Delta b(j))m_i \leq d(j)(q+1)^\alpha$ 
6      $p = d(j)$ ;
7     while  $(p \geq 0)$  do
8        $\delta = p(q+1)^\alpha + (d(j) - p)(q)^\alpha$ ;
9       if  $(\delta \geq (1 + \beta\Delta b(j))m_i)$ 
10         $result = p(q+1) + (d(j) - p)q$ ;
11      end if
12       $p = p - 1$ ;
13    end while
14    return  $result$ ;

```

---

Since it is a lower bound calculation procedure, the result shows the least amount of QC hours required in the given circumstances. We can now calculate  $q$ . If  $q$  was allowed to be fractional we would need to solve it by (2.41).

$$d(j)\hat{q}^\alpha = (1 + \beta\Delta b(j))m_i \quad \Rightarrow \quad \hat{q}^\alpha = \frac{(1 + \beta\Delta b(j))m_i}{d(j)}$$

$$\hat{q} = \left\lceil \left( \frac{(1 + \beta\Delta b(j))m_i}{d(j)} \right)^{1/\alpha} \right\rceil \quad (2.41)$$

**Theorem:** The number of quay crane hours calculated using the Algorithm (5) is a lower bound on the number of QC hours needed to serve vessel  $i$  when berthed  $\Delta b$  units away from the desired position and having a stay of  $d$  time units at the port.

**Proof:**

See Appendix (A.1) for proof  $\square$

**Corollary:** There is always a QC assignment plan which only includes  $\hat{q}$  or  $\hat{q} + 1$  number of QCs for each period (in which vessel  $i$  is at port), and this plan satisfies total QC requirement of vessel  $i$  (i.e.  $(1 + \Delta b\beta)m_i$ ) and minimizes the

total number of QC hours needed to serve vessel  $i$  when berthed  $\Delta b$  units away from the desired position.

**Proof:**

Proven Theorem guarantees Corollary, see Appendix (A.1) for proof of Theorem  $\square$

#### 2.4.4.3 Probing-1: Feasible assignment set fixing

We classify the next two methods as *probing methods* since they fix the value of one variable to one and analyze the immediate consequences. If as a consequence the lower bound rises above the upper bound then the corresponding variable can be eliminated. These methods are explained using the notation for the time-variant GSPP, but they work just as well for the time-invariant version.

The procedure goes through all variables  $y_j$  and iteratively fixes them to one. Fixing a variable to one usually implies that many other variables (columns) are becoming infeasible due to the overlap in berth/time space. Let  $I(j)$  be the set of infeasible columns when column  $j$  is used. A lower bound for the total cost when having selected  $j$  is:

$$\underline{z}^3(j) = \phi(j) + \sum_{i \in V \setminus \tau(j)} \left( \min_{j' \in \Omega(i) \setminus I(j)} \{\phi(j')\} \right) \quad (2.42)$$

the formula uses the cost lower bound of column  $j$  and adds the best cost of the remaining ships' columns. Taking into account that columns infeasible with the selection of column  $j$  are not included in the calculation, we have that if  $\underline{z}^3(j)$  turns out to be greater than  $\bar{z}$  then column  $j$  can be removed.

#### 2.4.4.4 Probing-2: Vessel pairs fixing

The second probing method extends the previous method by considering pairs of vessels when computing lower bounds. The method starts by pairing vessels. First the variable that assigns ship  $j$  with lowest  $\phi(j)$  is found. Among these  $N$  assignments the ones that overlap the most in time/berth space are selected to form the first pair. These assignments are removed from the set of available assignments and another pair is formed by selecting the ones with most overlap among the remaining assignments. This continues until all vessels are paired up (or one vessel remains). Now that vessels have been paired up, we can compute

a lower bound on the contribution of each vessel pair. For a vessel pair  $\{i_1, i_2\}$  this is done by (2.43).

$$\underline{z}(i_1, i_2) = \min_{j_1 \in \Omega(i_1), j_2 \in \Omega(i_2) \setminus I(j_1)} \{\phi(j_1) + \phi(j_2)\} \quad (2.43)$$

i.e. by selecting an assignment  $j_1$  for vessel  $i_1$  and an assignment  $j_2$  for vessel  $i_2$  that minimizes  $\phi(j_1) + \phi(j_2)$  and do not overlap. Let  $\mathcal{P}$  be the set of pairs and assume  $N$  is even. A lower bound for the total objective is

$$\underline{z}^4 = \sum_{\{i_1, i_2\} \in \mathcal{P}} \underline{z}(i_1, i_2) \quad (2.44)$$

we again go through all  $j \in \Omega$  and fix  $y_j$  to one, iteratively. Fixing  $y_j$  to one has several effects. The ship  $\tau(j)$  corresponding to column  $j$  is part of exactly one pair from  $\mathcal{P}$ . Since  $j$  is fixed we may have to redo our choice for that pair. This amounts to finding the best assignment  $j'$  for the other ship in the pair while ensuring that assignments  $j$  and  $j'$  do not overlap. For the other pairs we check if the best assignment for that pair overlaps with  $j$ . If not, we go on and use the best assignment, if there is an overlap we search for the best assignment pair that does not overlap with  $j$ .

Let  $i$  be the ship that was paired up with  $\tau(j)$  then we can write the lower bound obtained by fixing  $y_j = 1$  formally as:

$$\begin{aligned} \underline{z}^4(j) = & \phi(j) + \min_{j' \in \Omega(i) \setminus I(j)} \{\phi(j')\} \\ & + \sum_{\{i_1, i_2\} \in \mathcal{P} \setminus \{i, \tau(j)\}} \left( \min_{j_1 \in \Omega(i_1) \setminus I(j), j_2 \in \Omega(i_2) \setminus (I(j) \cup I(j_1))} \{\phi(j_1) + \phi(j_2)\} \right) \end{aligned} \quad (2.45)$$

We can eliminate  $y_j$  whenever  $\underline{z}^4(j) > \bar{z}$ . The two probing algorithms are rather time consuming, but the running time can be kept at a reasonable level by careful implementation. The two simpler preprocessing routines are also executed before running the probing methods in order to reduce the set of available columns.

#### 2.4.4.5 Valid inequality based on $\phi(j)$

The computed  $\phi(j)$  bounds give rise to a simple inequality that eliminates some non-optimal solutions from the solution space:

$$\sum_{j \in \Omega} \phi(j) y_j \leq \bar{z} \quad (2.46)$$

Constraint (2.46) ensures that the sum of all lower bounds of columns cannot be larger than the upper bound. The inequality can cut away feasible integer solutions, but only those that have an objective greater than the upper bound. It is likely that a black-box solver will be able to generate cover inequalities from the inequality since it is a knapsack constraint.

#### 2.4.4.6 Reduction of $r_{itq}$ variables

Finally, the time-variant version of GSPP may be improved with respect to variables  $r_{itq}$ . There can be no QC assignment before the EST of vessels and assignment of QCs have to be within the given interval  $R_i = [r_i^{min}, r_i^{max}]$ . Constraints (2.47) and (2.48) eliminate QC assignments that do not satisfy these requirements.

$$r_{itq} = 0 \quad \forall i \in V, q \notin R_i, t \in T \quad (2.47)$$

and

$$r_{itq} = 0 \quad \forall i \in V, q \in R_i, t \in T : t < EST_i \quad (2.48)$$

The preprocessing and probing described above remove some  $y_j$  variables. This can force some  $r_{itq}$  to zero. We let it be up to the preprocessing routines of the black-box IP solver to eliminate such  $r_{itq}$  variables.

### 2.4.5 Discussion of solution methods

In this paper we generate the complete models (2.23-2.27 for the time-invariant case and 2.28-2.35 for the time-variant case) using the columns that are left after the column reduction techniques presented in Section 2.4.4. These models are then solved by CPLEX.

Since the models contain a large number of columns, an alternative solution approach would be to solve the LP relaxation of the models using a column generation algorithm and obtain integer solutions using a branch-and-price algorithm. Such an approach has been used for related problems by, for example, Vacca et al. (2013) and Robenek et al. (2014). A simpler alternative to branch-and-price is to solve the integer model in the last iteration using the columns generated while solving the LP relaxation of the complete model using a column

generation algorithm. Such an approach is used by Saadaoui et al. (2015), but it is not guaranteed to obtain an optimal solution when the problem is solved in this way.

It is not clear if a branch-and-price approach would be advantageous for the size of BACAP instances currently used in the literature (see e.g. Meisel and Bierwirth (2009)) since CPLEX in general is very good at solving GSPP as long as the model fits into memory. However for larger instances branch-and-price algorithms will be competitive considering that at some point the number of generated columns for the complete models simply becomes too large to fit in the memory (see Saadaoui et al. (2015), for evidence of this for the BAP). The most interesting research direction related to branch-and-price algorithms is perhaps to use model 2.23-2.27 to solve the time-variant case (see comments at the end of Section 2.4.1) since we expect that the LP relaxation of the time-variant version of model 2.23-2.27 would be tighter than that of 2.28-2.35. Generating all columns for this model variant (2.23-2.27) is out of the question for all but the smallest instances, but its LP relaxation could be solved using column generation and a branch-and-price algorithm would therefore be feasible.

## 2.5 Computational Results

We compare our results to those that can be obtained by the model presented by Meisel and Bierwirth (2009). All models are solved by using the CPLEX 12.6 solver. The column generator and reduction techniques are implemented in C++. In order to have a fair comparison, the model presented by Meisel and Bierwirth (2009) is run for 10 hours using our computer and CPLEX version. The best result from the original and our re-implementation are reported. All tests are run on a 32 core AMD Opteron at 2.8Ghz and 132Gb of RAM. All running times are measured in seconds. The running times are reported for both the column generation and solver times. Due to memory restrictions only 5 threads are active per experiment.

### 2.5.1 Benchmark instances and running conditions

The benchmark is provided by Meisel and Bierwirth (2009). The data set includes three main vessel types (namely Feeders, Medium, and Jumbo vessels). Furthermore each vessel type differs in technical specifications and cost values. The generation of these instances is based on empirical data. The benchmark consists of 30 instances, and contains ten instances of 20, 30, and 40 vessels,



respectively. We consider a container terminal with a quay of length  $L=1000$  meters with 10 QCs available. The planning horizon is one week (168 hours), and planning operations are based on working hours. It should be noted that the planning horizon is set as a hard constraint by the benchmarks.

The vessels' specifications and parameters regarding the arrival and finishing time and the cost values can be obtained from Meisel and Bierwirth (2009). The interference coefficient ( $\alpha$ ) is set to 0.9, and the increase in the QC-hours needed due to berthing deviation is set to 0.01 (Meisel and Bierwirth (2009)).

The complete column generation procedure works as follows: First, all feasible columns are generated, and after that the two preprocessing techniques are applied. The probing methods described in Section 2.4.4.4 are run last, since they are the most time consuming and therefore it is beneficial to reduce the set of columns as much as possible before running them.

The models based on Meisel and Bierwirth (2009) (time-variant and time-invariant versions) are rerun with the same conditions reported in their paper. CPLEX 12.6 is run with a time limit of 36000 seconds using the options: *emphasize optimality* and *aggressive cut generation*. It is observed that the compact models require the *aggressive cut generation* option since the computation for the root node relaxation takes only little time, and most of the time is spent on branching for an integer solution.

For the set partitioning formulations, we observed that the best results were obtained by setting the MIP emphasis parameter to *discover hidden feasible solutions* and by turning on *local branching heuristic*. Another strategy that is applied to overcome the problem of finding an integer initial solution is to warm-start the models. Such solutions are also necessary for providing an upper bound for the column reduction strategies. Section 2.5.2 explains how the warm start solutions are found.

## **2.5.2 Upper bound and warm start strategies**

Warm starts (and consequently the upper bounds) are obtained by solving a simpler version of the GSPP model. For the time-invariant GSPP, modeling each berth with a length of 4 units provides a warm start for all versions of time-invariant models (berth length of 1 (BL=1), berth length of 2 (BL=2), and dynamic discretization). This simpler model is solved with a time limit of 15 minutes and, in most cases, the model can be solved to optimality.

For the time-variant version one can use the solution from the time-invariant

model as long as the discretization policy is kept the same, e.g. the solution to the time-invariant GSPP (BL=1) is not an upper bound for the dynamic-discretized time-variant GSPP. The time-invariant version of the model, which will generate a warm start, is run with a time limit of 20 minutes (not including the time to obtain BL=4 results). The additional runtime depends on the computational time generating the upper bound of the time-invariant case. In small and medium scale instances, the upper bounds obtained from the time-invariant models perform quite well. However, for large scale instances, BL=1 time-invariant models do not perform well. Hence, for the BL=1 versions the upper bound is selected among the time-invariant version with a berth length of 2 or dynamic discretization.

### 2.5.3 Computational Results

We present results for both the time-invariant and time-variant versions of the GSPP models and compare the results with those provided by the model of Meisel and Bierwirth (2009). Moreover we analyze the impact of the three berth discretizations and the column reduction techniques.

The performance of GSPP models are presented in Tables 2.4-to-2.7. In each table, the first column, "#", indicates instance ID. The columns denoted "Z" show the best upper bounds obtained, while "LB" reports the best lower bounds found. The gap ( $G$ ) is calculated between upper and lower bounds. In Tables 2.4 and 2.7, the " $T_C$ " and " $T_{OPT}$ " are the time spent (in seconds) generating columns and the time spent solving the mathematical model, respectively. The column " $R_+$ " illustrates whether the optimal solution is found in the root node relaxation. If yes, there is a "+", otherwise a "-". The column  $R_{LB}$  reports the lower bounds obtained in the root node. It should be noted that if the instance is solved to optimality in the root node, there is no root node LB. Additionally, the number of nodes in the branch and bound (B&B) tree is presented in "#nodes B&B" columns. The following four columns show the effect of the column reduction techniques. The column under column reduction section named  $|\Omega|$  shows the number of columns generated. After that, the number of columns left is reported in each cell. Column  $|\Omega_1|$  shows the number of columns left after the two simple preprocessing steps, while  $|\Omega_2|$  and  $|\Omega_3|$ , shows the number of columns left after the probing methods 1 and 2, respectively. The two columns ("UB  $T_{OPT}$ " and " $Z_{UB}$ ") report the upper bound used as a warm start solution and the length of time we spent computing this upper bound. Different from Table 2.7, Table 2.4 (for the time-invariant case) contains results that show the performance without using the preprocessing steps in the last four columns.

### 2.5.3.1 Time-Invariant GSPP results

The computational results for the time-invariant version of the GSPP are shown in Table 2.4 and 4.3. There is a clear tradeoff between the number of columns and solution quality.

Table 2.4 shows the results for a berth length of 1 unit ( $l_s$  : 10m in real-life, BL=1) which is the original problem studied in Meisel (2009e). In this case, the GSPP formulation produces optimal results for all small and medium scale instances ( $N = 20, 30$  vessels). For large scale instances ( $N = 40$  vessels), only four instances cannot be solved to optimality within the 10 hour time limit. For small and medium scale instances, the runtime ( $T_{OPT}$ ) is always less than 13 minutes, and the optimal solution is often found in the root node. This is clearly not the case for the instances with large scale instances.

Note that the column generation time ( $T_C$ ) is small for any type of instance. In most small and medium scale instances, it is less than 10 seconds. The number of generated columns can be reduced significantly using the proposed rules. For small scale instances, simple preprocessing can, on average, reduce 77% of the columns, while for medium and large scale instances the reduction drops to 58% and 20% respectively. The main reason for the drop in effectiveness is the quality of upper bounds obtained for each class of instance and the complexity of the analyzed instances. The results also reveal that the second probing algorithm is more effective than the first, and can reduce the number of columns even further. In total 85% of all columns are removed on average in the small scale instances, while for medium and large scale instances the number is 70% and 28% on average, respectively. The upper bounds computed initially (as warm start) are relatively tight and only two upper bounding models cannot be solved to optimality within 15 minutes. There is an average of 7% of optimality gap (see  $G_{UB}$  column for each instance size in Table 2.4). Model with small time limits also outperforms SWO heuristic for 8 instances of 10 large scale instances (see SWO ( $Z_H$ )<sub>best</sub> column in Table 2.5). These results show that the performance of GSPP formulations for small time limits is also strong. The results of GSPP (BL=1) without any probing (but including the two simple preprocessing methods) are presented in the last columns of Table 2.4. For this experiment there is no clear winner, but one can argue that the extra complexity involved in the probing algorithms does not pay off here.

Table 2.5 summarizes results for the time-invariant BACAP. For this problem, the known upper and lower bounds have been improved for all instances. The GSPP (BL=1) results outperform BL=2 and dynamic discretization results for all instances except #21, #23. For instance #23, BL=2 and dynamic discretization policies both present best upper bound, while for instance #21, dynamic

discretization performs the best. The dynamic discretization policy still finds the optimum solutions of original problem (BL=1) for 14 out of 30 instances. We can conclude that finer discretization outperforms other discretization methods for most of the instances. The last column of Table 2.5 presents the results of squeaky wheel optimization (SWO) heuristic which is proposed by Meisel (2009e) for BACAP with time-invariant QC assignment (BL=1 version).

**Table 2.5:** Results of GSPP, SWO heuristic-BACAP-TI (Meisel (2009e))

N	#	GSPP (BL=1)			GSPP (BL=2)			GSPP (Dynamic)			SWO
		Z	LB	G	Z	LB	G	Z	LB	G	$(Z_H)_{best}$
20	1	89.0	89.0	0.0%	92.8	92.8	0.0%	91.4	91.4	0.0%	89.0
	2	56.2	56.2	0.0%	56.2	56.2	0.0%	56.2	56.2	0.0%	56.2
	3	85.7	85.7	0.0%	87.7	87.7	0.0%	85.7	85.7	0.0%	89.8
	4	81.8	81.8	0.0%	94.4	94.4	0.0%	81.8	81.8	0.0%	81.8
	5	59.2	59.2	0.0%	60.4	60.4	0.0%	60.4	60.4	0.0%	59.2
	6	59.2	59.2	0.0%	62.8	62.8	0.0%	59.2	59.2	0.0%	59.2
	7	75.2	75.2	0.0%	77.0	77.0	0.0%	77.0	77.0	0.0%	75.8
	8	61.4	61.4	0.0%	68.4	68.4	0.0%	61.4	61.4	0.0%	61.4
	9	79.0	79.0	0.0%	79.0	79.0	0.0%	79.0	79.0	0.0%	79.0
	10	101.0	101.0	0.0%	102.5	102.5	0.0%	102.0	102.0	0.0%	105.0
		$G_{av}$		0.0%	$G_{av}$		0.0%	$G_{av}$		0.0%	
		$(T_{opt})_{av}$		126	$(T_{opt})_{av}$		36	$(T_{opt})_{av}$		76	
30	11	143.2	143.2	0.0%	143.2	143.2	0.0%	143.2	143.2	0.0%	148.0
	12	92.0	92.0	0.0%	92.0	92.0	0.0%	92.0	92.0	0.0%	93.4
	13	110.0	110.0	0.0%	110.0	110.0	0.0%	110.0	110.0	0.0%	111.2
	14	107.4	107.4	0.0%	112.4	112.4	0.0%	107.6	107.6	0.0%	115.8
	15	168.4	168.4	0.0%	170.8	170.8	0.0%	170.8	170.8	0.0%	175.8
	16	121.6	121.6	0.0%	121.8	121.8	0.0%	121.6	121.6	0.0%	126.6
	17	109.4	109.4	0.0%	109.4	109.4	0.0%	109.4	109.4	0.0%	114.6
	18	135.0	135.0	0.0%	143.0	143.0	0.0%	136.2	136.2	0.0%	144.4
	19	176.2	176.2	0.0%	179.8	179.8	0.0%	176.2	176.2	0.0%	180.8
	20	139.8	139.8	0.0%	141.0	141.0	0.0%	139.8	139.8	0.0%	139.8
		$G_{av}$		0.0%	$G_{av}$		0.0%	$G_{av}$		0.0%	
		$(T_{opt})_{av}$		410	$(T_{opt})_{av}$		107	$(T_{opt})_{av}$		234	
40	21	247.0	233.1	5.6%	250.6	250.6	0.0%	246.8	246.8	0.0%	293.2
	22	178.4	178.4	0.0%	185.4	185.4	0.0%	180.6	180.6	0.0%	193.8
	23	269.0	247.8	7.7%	268.6	268.6	0.0%	268.6	264.2	1.7%	331.4
	24	307.0	283.8	7.6%	310.6	303.8	2.2%	307.0	292.5	4.7%	366.0
	25	164.6	164.6	0.0%	168.2	168.2	0.0%	165.8	165.8	0.0%	171.6
	26	258.2	258.2	0.0%	265.2	265.2	0.0%	258.2	258.2	0.0%	278.4
	27	205.4	205.4	0.0%	216.8	216.8	0.0%	206.6	206.6	0.0%	235.8
	28	300.1	267.5	10.9%	314.8	282.6	10.2%	313.1	272.1	13.1%	412.4
	29	227.6	227.6	0.0%	235.2	235.2	0.0%	229.4	229.4	0.0%	255
	30	210.1	210.1	0.0%	224.4	224.4	0.0%	217.0	217.0	0.0%	284.2
		$G_{av}$		3.2%	$G_{av}$		1.2%	$G_{av}$		1.9%	
		$(T_{opt})_{av}$		21032	$(T_{opt})_{av}$		11565	$(T_{opt})_{av}$		14160	

The performance of the GSPP (BL=1) can be compared with a modified version of the model by Meisel and Bierwirth (2009) (presented in Section 2.3.2). Table 4.3 shows the results from this model. Upper and lower bounds, gaps, computational times, and the number of nodes in the B&B tree are presented.

ht

**Table 2.4:** GSPP (BerthLength=1, Fixed QC)-Original Problem with Fixed QC number

Computational Results of GSPP										Performance of Column Reduction					UB Generator		Results with probing disabled		
#	Z	LB	$G_1$	$T_C$	$ToPT$	$R_+$	$R_{UB}$	#Nodes B&B	$ \Omega $	$ \Omega_1 $	$ \Omega_2 $	$ \Omega_3 $	$ToPT$	UB	$Z_{UB}$	Z	LB	$G_2$	$ToPT$
1	89.0	89.0	0.0%	2	142	+	-	-	316278	102471	88598	62677	21	97.4	89.0	89.0	0.0%	90	
2	56.2	56.2	0.0%	<1	<1	+	-	-	338438	19133	15785	687	27	57.4	56.2	56.2	0.0%	26	
3	85.7	85.7	0.0%	1	186	+	-	-	252618	94083	84536	71287	28	88.9	85.7	85.7	0.0%	116	
4	81.8	81.8	0.0%	1	160	+	-	-	390624	106970	99809	79138	47	94.4	81.8	81.8	0.0%	148	
5	59.2	59.2	0.0%	1	31	+	-	-	275238	29887	24487	20471	15	61.6	59.2	59.2	0.0%	23	
6	59.2	59.2	0.0%	<1	7	+	-	-	305485	27446	21672	11544	25	67.6	59.2	59.2	0.0%	54	
7	75.2	75.2	0.0%	1	59	+	-	-	330639	58441	51442	26434	26	77.6	75.2	75.2	0.0%	161	
8	61.4	61.4	0.0%	1	100	+	-	-	359703	61200	54227	48862	31	72.2	61.4	61.4	0.0%	117	
9	79.0	79.0	0.0%	1	152	+	-	-	286683	91399	84533	66469	27	89.0	79.0	79.0	0.0%	179	
10	101.0	101.0	0.0%	2	282	-	97.6	1480	359320	132706	126499	115110	48	106.1	101.0	101.0	0.0%	349	
		$G_{av}$	0.0%						$R_{av}$	77%	12%	31%		$G_{UB}$	9%				
11	143.2	143.2	0.0%	6	418	+	-	-	496340	250999	231616	202530	63	151.4	143.2	143.2	0.0%	506	
12	92.0	92.0	0.0%	1	84	+	-	-	523233	100586	88348	55483	36	95.0	92.0	92.0	0.0%	140	
13	110.0	110.0	0.0%	2	143	+	-	-	483010	122017	110899	95635	25	112.8	110.0	110.0	0.0%	141	
14	107.4	107.4	0.0%	3	161	+	-	-	467661	164949	154643	112752	30	117.2	107.4	107.4	0.0%	244	
15	168.4	168.4	0.0%	5	621	+	-	-	477733	283078	273274	248677	76	181.8	168.4	168.4	0.0%	753	
16	121.6	121.6	0.0%	3	169	+	-	-	532026	181721	167068	102007	71	122.8	121.6	121.6	0.0%	376	
17	109.4	109.4	0.0%	3	119	+	-	-	467085	142579	130493	92707	45	114.0	109.4	109.4	0.0%	134	
18	135.0	135.0	0.0%	8	556	-	131.9	158	508300	251673	240444	233079	77	150.2	135.0	135.0	0.0%	465	
19	176.2	176.2	0.0%	15	748	-	174.3	3259	506300	312026	302952	272114	70	189.4	176.2	176.2	0.0%	890	
20	139.8	139.8	0.0%	11	475	+	-	-	487945	238218	224054	163891	63	145.8	139.8	139.8	0.0%	451	
		$G_{av}$	0.0%						$R_{av}$	58%	7%	20%		$G_{UB}$	6%				
21	247.0	233.1	5.6%	11	36000	-	220.9	117418	768104	614655	605174	577530	290	253.2	246.8	242.8	1.6%	36000	
22	178.4	178.4	0.0%	10	532	+	-	-	611023	403976	386376	295991	83	189.0	178.4	178.4	0.0%	821	
23	269.0	247.8	7.7%	21	36000	-	242.0	34137	720932	616879	608838	589421	417	281.6	268.0	247.4	7.7%	36000	
24	307.0	283.8	7.6%	32	36000	-	272.7	36504	657415	634437	625994	601113	900	326.8	309.8	283.6	8.4%	36000	
25	164.6	164.6	0.0%	7	1012	-	138.8	4953	553500	338534	338568	309849	88	176.4	164.6	164.6	0.0%	1277	
26	258.2	258.2	0.0%	23	31523	-	236.4	118708	675819	596031	586485	552397	879	278.9	258.2	258.2	0.0%	33669	
27	205.4	205.4	0.0%	10	1206	-	198.8	2909	615092	474731	459672	427398	85	220.2	205.4	205.4	0.0%	1523	
28	300.1	267.5	10.9%	35	36000	-	256.4	27200	698783	631227	627241	612621	900	320.2	299.5	270.6	9.7%	36000	
29	227.6	227.6	0.0%	26	8077	-	212.1	34227	673090	501858	493100	454074	132	241.4	227.6	227.6	0.0%	3629	
30	210.1	210.1	0.0%	31	20966	-	196.0	108339	678072	523970	508499	463573	559	235.6	210.1	210.1	0.0%	25400	
		$G_{av}$	3.2%						$R_{av}$	20%	2%	7%		$G_{UB}$	7%				

$$G_1 = \left\lceil \frac{Z-L_B}{Z} \right\rceil, G_2 = \left\lceil \frac{Z_{UB} - \text{Probe} - L_{B_{UB}} - \text{Probe}}{Z_{UB} - \text{Probe}} \right\rceil, (R_{av})_i = \left\lceil \frac{|\Omega_1| - |\Omega_2|}{|\Omega|} \right\rceil, G_{UB} = \left\lceil \frac{Z_{UB} - Z_{UB} - \text{Probe}}{Z_{UB}} \right\rceil, G_{av} = \left\lceil \frac{\sum_n G_1}{n} \right\rceil.$$

Results show that even for the small scale instances, there are two cases in which no integer solution was found. In this benchmark, only five instances are solved to optimality. The GSPP formulation solves all the instances to optimality in much shorter times. For medium and large scale instances, there are only two instances in which an upper bound is obtained and the lower bounds are significantly worse than those from the GSPP formulation.

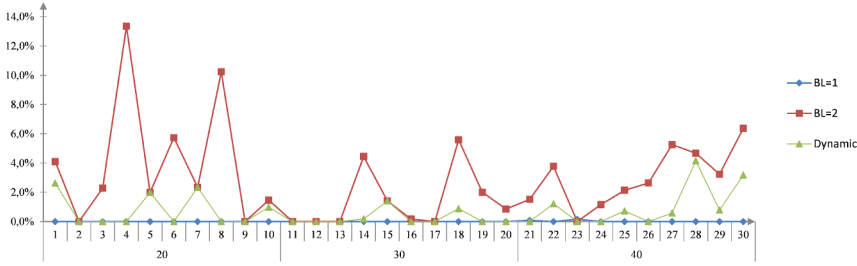
**Table 2.6:** Reformulation of compact model-BACAP-TI

#	Z	LB	G	$T_{OPT}$	#Nodes	#	Z	LB	G	#Nodes	#	Z	LB	#Nodes
1*	89.0	86.2	3.1%	*	2769450	11*	150.4	108.3	27.9%	1557183	21*	X	116.4	762709
2	56.2	56.2	0.0%	7397	383896	12*	X	74.1	-	1205386	22*	X	97.5	438579
3*	X	67.6	-	*	2605595	13*	X	88.3	-	1408812	23*	X	142.5	745119
4*	X	68.7	-	*	2166910	14*	X	85.3	-	1284386	24*	X	139.4	415411
5	59.2	59.2	0.0%	4151	466141	15*	X	97.3	-	1427223	25*	X	115.9	824894
6	59.2	59.2	0.0%	105	5428	16*	X	96.2	-	1500854	26*	X	102.4	433046
7*	75.2	75.1	0.16%	*	1748332	17*	X	87.9	-	1311279	27*	X	133.4	947127
8	61.4	61.4	0.0%	12057	876111	18*	135.7	106.5	21.5%	1694085	28*	X	140.8	721192
9	79.0	79.0	0.0%	4464	385267	19*	X	118.0	-	1019351	29*	X	123.0	792108
10*	101.0	90.5	10.4%	*	4744212	20*	X	105.1	-	1135501	30*	X	128.6	905548

$G = \lfloor \frac{Z-LB}{Z} \rfloor$ , \* represents that 10h time-limit has been reached

Finally, the performance of dynamic discretization which shows promising results is evaluated in detail. Dynamic discretization models have less columns than the BL=1 case, but more than the BL=2. Since discretization is one in 5 units around the desired berthing position, and two for the rest, the number of columns is closer to the GSPP (BL=2). On average, the number of columns is 44% less than the BL=1 case. For small and medium instances, the computational time needed to reach optimality is reduced by 50%, and 54% compared to BL=1 case ( $T_{GAP} = \frac{T_{OPT,Dynamic} - T_{OPT,BL=1}}{T_{OPT,BL=1}}$ ). The results show that, with dynamic discretization, optimal solutions can be obtained for all but 3 instances (#23, #24, #28). It is observed that for instances (#21, #23), which are not solved to optimality in the BL=1, dynamic discretization obtains a lower objective value in 10 hours of computational time.

Figure 2.3 illustrates the gap between each discretization policy and the best known solution for each instance. Apart from instance (#21, #23), BL=1 discretization presents the best upper bounds for each instance. Dynamic discretization performs better than BL=2 case in all instances (in some instance they found identical solutions).



**Figure 2.3:** Gap (%) from best known solution for BL=1, BL=2 and dynamic discretized BACAP- Time-Invariant QC policy

### 2.5.3.2 Time-Variant GSPP results

Tables 2.7 and 2.8 present the results of the time-variant version of the GSPP formulation (see, Section 2.4.2). As before, we first present results for a berth length of 1 unit. Afterwards, we present summary of results for different discretization policies and solution approaches.

The results from the BL=1 case are reported in Table 2.7. This problem corresponds to the one solved by Meisel and Bierwirth (2009). The results in Table 2.7 show that only one of the small instances cannot be solved to optimality. For medium instances, the average gap is less than 2%, while for large instances it is around 15%. It should be noted that the standard deviation of the optimality gap for large instances is high.

The column reduction techniques remove a larger fraction of the columns compared to the time-invariant case, but the number of surviving columns has nevertheless increased by a factor of 2.8 on average. Overall, all four reduction tools have deleted 90% of the columns in the small instances, while 72% and 27% are reduced for medium and large scale instances, respectively. The computational time to generate the columns remains acceptable for small and medium scale instances, for large scale instances the average time is 10 minutes.

Compared to the time-invariant case the complexity of the model has increased and it becomes reasonable to see that the quality of the warm start solutions have decreased. For several large scale instances, the model faces memory issues. Six of them ran out of memory and were rerun using only one thread (which reduces memory usage).

Table 2.7: GSPP (BerthLength=1, Variable QC assignment)

Computational results of GSPP										Performance of column reduction				UB Generation	
#	Z	LB	$G_1$	$T_C$	$T_{OPT}$	$R_+^*$	$R_{LB}$	#Nodes B&B	$ \Omega $	$ \Omega_1 $	$ \Omega_2 $	$ \Omega_3 $	UB $T_{OPT}$	$Z_{UB}$	
1	84.1	84.1	0.0%	20	321	-	83.3	2887	1234646	317414	261017	159035	142	89.0	
2	53.9	53.9	0.0%	1	2	+	-	-	1086872	68846	57230	4265	1	56.2	
3	76.3	76.3	0.0%	4	1096	-	75.0	1081145	936469	321403	279470	242882	186	85.7	
4	76.2	76.2	0.0%	3	717	-	74.0	82907	1366755	255422	233926	153491	160	81.8	
5	56.8	56.8	0.0%	<1	56	+	-	-	882054	87858	71329	54527	31	59.2	
6	57.6	57.6	0.0%	<1	1	+	-	-	1103729	23934	16177	1411	7	59.2	
7	68.0	68.0	0.0%	7	203	-	67.1	2614	1197949	210198	184265	85013	59	75.2	
8	56.1	56.1	0.0%	1	87	-	55.9	45	1193809	111140	89978	71392	100	61.4	
9	75.1	75.1	0.0%	5	924	-	73.2	72287	954820	240732	213842	140247	152	79.0	
10	89.3	88.4	1.1%	19	36000	-	83.1	813843	1281315	464115	440292	391410	282	101.0	
		$G_{av}$	0.1%						$R_{av}$	82%	16%	43%	$G_{UB}$	8%	
11	139.8	139.1	0.5%	58	36000	-	133.8	219068	1744848	816725	744802	616533	418	143.2	
12	81.8	81.8	0.0%	3	337	+	-	-	1607596	341504	302519	189863	84	92.0	
13	102.4	102.4	0.0%	11	1122	-	100.5	10488	1727182	429088	381602	334247	143	110.0	
14	99.1	99.1	0.0%	23	12960	-	97.4	5798205	1525995	475467	440336	281220	161	107.4	
15	154.2	145.1	5.9%	27	36000	-	136.8	30374	1761414	995156	951700	830382	621	168.4	
16	115.4	111.6	3.3%	29	36000	-	107.8	330293	1796509	670045	618100	381185	169	121.6	
17	102.6	102.6	0.0%	20	1416	-	100.7	41326	1696500	474650	429690	293693	119	109.4	
18	121.9	120.0	1.6%	69	36000	-	117.4	804373	1709858	726816	677637	634660	556	135.0	
19	165.2	161.6	2.2%	349	36000	-	158.8	56899	2004197	1195939	1153256	1001574	748	176.2	
20	132.5	129.6	2.2%	243	36000	-	128.6	144075	1801078	812503	750895	521506	475	139.8	
		$G_{av}$	1.6%						$R_{av}$	61%	8%	24%	$G_{UB}$	8%	
21†	223.1	178.3	20.1%	51	36000	-	178.0	326	2513805	2056806	2023788	1928308	1200	250.6	
22	171.0	166.1	2.9%	445	36000	-	163.1	37143	2010843	1370334	1310042	963164	239	185.4	
23†	264.4	196.7	25.6%	335	36000	-	196.4	27	2364293	2014422	1983473	1923115	1200	269.2	
24†	302.2	245.4	18.8%	846	36000	-	239.4	22	2258599	2176590	2144639	2065263	1200	320.4	
25	148.1	141.2	4.7%	52	36000	-	139.9	10063	1690236	1117012	1041771	943454	195	168.2	
26†	245.0	208.8	14.8%	1327	36000	-	208.0	300	2269042	2005944	1965226	1831379	1200	265.2	
27	184.9	176.0	4.8%	113	36000	-	173.9	5507	2017218	1612962	1560262	1460709	423	216.8	
28†	315.2	215.3	31.7%	1499	36000	-	215.3	15	2543836	2340412	2327678	2277105	1200	320.2	
29	226.7	186.0	18.0%	433	36000	-	185.5	974	2163279	1669144	1643406	1495597	350	235.2	
30†	187.8	175.0	6.8%	477	36000	-	173.9	4103	2441545	1875349	181354	1614804	1200	226.7	
		$G_{av}$	14.8%						$R_{av}$	19%	3%	8%	$G_{UB}$	12%	

† represents that CPLEX solver has faced a memory overflow while the instance was running using 5 threads. Hence, † instances are rerun with only one thread. ‡ represents instances that run with one thread, and again faced a memory problem. The results are reported on the time when CPLEX ran out of memory.



We can conclude that the GSPP model can solve the problem considered in Meisel and Bierwirth (2009) to optimality, or near optimality, for instances with 20 or 30 vessels. For larger instances the performance is more erratic and only a subset of those instances can be solved reasonably well. In the following the performance of the time-variant GSPP model is compared with the compact model proposed by Meisel and Bierwirth (2009). Table 2.8 summarizes the results. In addition to the upper and lower bounds of the compact model, the best results from the heuristic procedures described in Meisel and Bierwirth (2009) are presented. Additionally, the results from the time-variant GSPP model and the Meisel and Bierwirth (2009) model with warm-start solutions are presented in Table 2.8. Results show that there are only four instances which are solved to optimality by the compact model, while 13 instances are solved to optimality by the BL=1 discretized GSPP model. The compact model performs very poorly on the medium and large scale instances in terms of producing upper bounds, and the lower bounds are consistently outperformed by the GSPP model. When warm-starts are imposed on Meisel and Bierwirth (2009) model, eight instances are solved to optimality, and four instances resulted in a better upper bound compared to the GSPP models. We also observe that the upper bounds produced by all versions of the GSPP models often improve upon the best known upper bounds produced by the state-of-the-art heuristics. We hope that the improved upper and lower bounds will be helpful in evaluating future heuristics and exact methods for the problem.

**Table 2.8:** Results of Model, heuristic results-BACAP-TV (Meisel and Bierwirth (2009)), GSPP results

N	#	Compact Model			Compact Model with warmstart			GSPP (BL=1)			GSPP (Dynamic)			GSPP (BL=2)			Heuristic ( $Z_H$ ) <sub>best</sub>
		Z	LB	G	Z	LB	G	Z	LB	G	Z	LB	G	Z	LB	G	
20	1	84.1	84.0	0.1%	84.1	84.1	0.0%	84.1	84.1	0.0%	86.1	86.1	0.0%	87.3	87.3	0.0%	85.1
	2	53.9	53.9	0.0%	53.9	53.9	0.0%	53.9	53.9	0.0%	53.9	53.9	0.0%	54.5	54.5	0.0%	53.9
	3	77.4	75.2	2.9%	76.3	76.1	0.3%	76.3	76.3	0.0%	77.4	77.4	0.0%	77.6	77.6	0.0%	77.4
	4	76.2	75.8	0.5%	76.2	75.9	0.4%	76.2	76.2	0.0%	77.9	77.9	0.0%	78.5	78.5	0.0%	77.9
	5	56.8	56.8	0.0%	56.8	56.8	0.0%	56.8	56.8	0.0%	58.1	58.1	0.0%	58.2	58.2	0.0%	56.8
	6	57.6	57.6	0.0%	57.6	57.6	0.0%	57.6	57.6	0.0%	57.6	57.6	0.0%	60.8	60.8	0.0%	57.6
	7	68.0	67.5	0.7%	68.0	68.0	0.0%	68.0	68.0	0.0%	69.3	69.3	0.0%	69.5	69.5	0.0%	68.9
	8	56.1	56.1	0.0%	56.1	56.1	0.0%	56.1	56.1	0.0%	57.2	57.2	0.0%	60.5	60.5	0.0%	56.1
	9	75.1	75.0	0.1%	75.1	75.1	0.0%	75.1	75.1	0.0%	75.3	75.3	0.0%	75.4	75.4	0.0%	75.5
	10	88.2	88.2	3.0%	89.3	88.9	0.4%	89.3	88.4	1.1%	89.4	88.4	1.1%	89.8	89.8	0.0%	93.0
30		$G_{av}$	$G_{av}$	0.7%	$G_{av}$	$G_{av}$	0.1%	$G_{av}$	$G_{av}$	0.1%	$G_{av}$	$G_{av}$	0.1%	$G_{av}$	$G_{av}$	0.0%	
		$(T_{opt})_{av}$	$(T_{opt})_{av}$	-	$(T_{opt})_{av}$	$(T_{opt})_{av}$	11643	$(T_{opt})_{av}$	$(T_{opt})_{av}$	3939	$(T_{opt})_{av}$	$(T_{opt})_{av}$	5983	$(T_{opt})_{av}$	$(T_{opt})_{av}$	2500	
	11	X	137.7	-	140.4	130.0	7.4%	139.8	139.1	0.5%	140.0	139.7	0.2%	140.5	140.5	0.0%	147.8
	12	81.8	81.4	0.4%	81.8	81.8	0.0%	81.8	81.8	0.0%	82.5	82.5	0.0%	82.7	82.7	0.0%	82.5
	13	104.9	100.9	3.9%	102.4	101.8	0.6%	102.4	102.4	0.0%	103.5	103.1	0.4%	104.8	103.7	1.1%	104.5
	14	X	96.8	-	99.1	98.7	0.4%	99.1	99.1	0.0%	102.0	101.8	0.2%	105.3	105.3	0.0%	105.8
	15	X	136.9	-	150.4	132.1	12.1%	154.2	145.1	5.9%	154.7	150.7	2.6%	154.9	152.9	1.3%	157.4
	16	X	106.2	-	113.8	109.1	4.1%	115.4	111.6	3.3%	113.8	112.8	0.9%	115.4	114.2	1.0%	118.5
	17	X	99.6	-	102.6	101.9	0.6%	102.6	102.6	0.0%	102.9	102.7	0.2%	103.2	103.2	0.0%	104.2
	18	X	117.8	-	121.9	118.8	2.5%	121.9	120.0	1.6%	122.8	120.9	1.5%	128.0	126.0	1.6%	125.5
40	19	X	156.4	-	165.2	146.1	11.5%	165.2	161.6	2.2%	166.4	165.1	0.8%	168.2	168.2	0.0%	173.8
	20	X	125.6	-	134.5	120.8	10.2%	132.5	129.6	2.2%	132.8	131.8	0.8%	133.1	132.7	0.3%	135.2
		$G_{av}$	$G_{av}$	2.2%	$G_{av}$	$G_{av}$	4.9%	$G_{av}$	$G_{av}$	1.5%	$G_{av}$	$G_{av}$	0.8%	$G_{av}$	$G_{av}$	0.5%	
		$(T_{opt})_{av}$	$(T_{opt})_{av}$	-	$(T_{opt})_{av}$	$(T_{opt})_{av}$	33950	$(T_{opt})_{av}$	$(T_{opt})_{av}$	23183	$(T_{opt})_{av}$	$(T_{opt})_{av}$	32418	$(T_{opt})_{av}$	$(T_{opt})_{av}$	21363	
	21	X	165.7	-	206.3	130.4	36.8%	223.1	178.3	20.1%	209.0	180.8	13.5%	205.0	184.6	9.9%	215.0
	22	X	159.6	-	170.9	145.4	14.9%	171.0	166.1	2.9%	171.2	168.7	1.5%	175.3	173.0	1.3%	178.8
	23	X	185.0	-	257.7	135.1	47.6%	264.4	196.7	25.6%	264.5	197.3	25.4%	257.7	199.6	22.5%	264.3
	24	X	224.1	-	288.9	184.1	36.3%	302.2	245.4	18.8%	303.8	246.2	19.0%	292.6	251.5	14.1%	326.6
	25	X	133.3	-	148.9	125.2	15.9%	148.1	141.2	4.7%	148.3	144.8	2.4%	151.8	149.5	1.5%	155.1
	26	X	201.3	-	238.7	152.1	36.3%	245.0	208.8	14.8%	245.2	211.4	13.8%	245.9	221.1	10.1%	259.6
50	27	X	172.2	-	183.9	148.7	19.1%	184.9	176.0	4.8%	183.9	178.6	2.9%	191.4	186.3	2.7%	200.8
	28	X	211.7	-	273.0	147.2	46.1%	315.2	215.3	31.7%	294.7	218.9	25.7%	256.5	220.1	14.2%	286.2
	29	X	180.3	-	214.1	146.4	31.6%	226.7	186.0	18.0%	211.9	187.9	11.3%	210.0	196.9	6.2%	219.4
	30	X	170.1	-	189.3	154.6	18.3%	187.8	175.0	6.8%	187.5	180.7	3.6%	193.4	186.4	3.6%	240.9
		$G_{av}$	$G_{av}$	-	$G_{av}$	$G_{av}$	30.2%	$G_{av}$	$G_{av}$	14.8%	$G_{av}$	$G_{av}$	11.9%	$G_{av}$	$G_{av}$	8.6%	
		$(T_{opt})_{av}$	$(T_{opt})_{av}$	-	$(T_{opt})_{av}$	$(T_{opt})_{av}$	36000	$(T_{opt})_{av}$	$(T_{opt})_{av}$	36000	$(T_{opt})_{av}$	$(T_{opt})_{av}$	36000	$(T_{opt})_{av}$	$(T_{opt})_{av}$	36000	

 $(Z_H)_{best}$ : The best solution of an instance for three heuristics (SWO, TS, FCFS with local refinements)

Table 2.8 also helps to compare performance of different discretization methods. Since BL=1 discretization solves many of small and medium scale instances to optimality, BL=1 outperforms dynamic and BL=2 discretization for all such instances except instance #16, for which dynamic discretization presents the best upper bound. For small and medium scale instances, dynamic discretization continuously outperforms BL=2. For large scale instances, comparison is more interesting. BL=1 discretization performs better for three out of ten instances. While, BL=2 outperforms the other two methods in five instances, and dynamic discretization does the same with two instances.

### 2.5.3.3 Results for the berth allocation problem

The GSPP model, tested until now, can also solve the BAP by simply fixing the QC assignment decisions. This would result in a model similar to the one presented by Buhrkal et al. (2011). The latter, however, has only been tested for the discrete variant of the problem, where each berth holds exactly one vessel. Here we show how such models perform on the continuous variant. This problem is a special case of the BACAP.

We changed the same set of instances to include the number of QCs assigned (and consequently the processing time) as a parameter. The required change in the benchmark is achieved by replacing  $r_i^{min}, r_i^{max}$  in each data set with  $\bar{r}_i = \left\lceil \frac{r_i^{min} + r_i^{max}}{2} \right\rceil$ . Hence, the generation of columns will be based on the single value of the  $\bar{r}_i$  parameter. The time-invariant version of GSPP without the knapsack constraints (constraint (2.26)) will be used for this experiment. The rest of the parameters are kept the same. We use the instances from Meisel and Bierwirth (2009) and solely test the BL=1 discretization. Column reduction techniques were not applied and no upper bounds were computed a-priori.

The results are summarized in Table 2.9. Each row corresponds to 10 instances, the first column reports the instance size, while the next shows the average optimality gap (all instances were solved to optimality). The next column shows the average time needed to generate the model, then follows the time for solving the IP model and the last column shows the average number of columns generated. For the two last columns the number in square brackets indicates the standard deviation ( $\sigma_N$ ) and coefficient of variation ( $\sigma_N/\mu$ ).

**Table 2.9:** Results for the BAP without quay crane decisions

$N$	$G_{av}$	$T_c$	$T_{OPT}$	$ \Omega $
20	0.00%	3	131 [40,0.30]	141391 [17472,0.12]
30	0.00%	5	340 [143,0.42]	215967 [9128,0.04]
40	0.00%	6	893 [541,0.60]	286668 [25173,0.08]

The results show that the GSPP model is able to handle the fine discretization very well and the application of the model to the standard BAP can be taken further than what was done by Buhrkal et al. (2011). It seems likely that even larger instances could be solved within a couple of hours, but we have not tested this. We also conducted tests with the BL=2 discretization and the dynamic discretization. For the 40 ship instances the average solution costs were 2.8% and 1.71% higher than those from the BL=1 discretized model, respectively. Meanwhile the average running time was 209 seconds for BL=2 models and 606 seconds for dynamic discretized models. Based on the results, we recommend using the BL=1 model for instances of this size or smaller.

## 2.6 Conclusions and suggestions for future work

In this study, we have proposed novel GSPP formulations for the BACAP considering both time-variant and time-invariant QC assignment policies. The proposed models solve the problem introduced in Meisel and Bierwirth (2009). Computational results show that the performances of both the time-variant and time-invariant GSPP formulations are strong with respect to both upper and lower bounds. In particular, the GSPP formulation can provide optimal solutions in relatively short computation times for the small and medium sized instances. For these instance sizes, all instances could be solved to optimality for the time-invariant case, while 13 out of 20 instances could be solved for the time-variant case. For large scale instances, the objective value and lower bounds have been improved. We believe that the improved bounds would be useful in the evaluation of new heuristics to solve such instances. Note that both upper and lower bounds have been improved compared to the state-of-the-art results for all 60 instances when the results of compact model with warm start is also taken into account, and for 56 of 60 instances otherwise. This chapter also discusses the effects of time-variant and time-invariant QC assignment policy for terminals. We show that there is an additional cost of time-invariant QC policy and we quantify this difference, although for artificial instances.

The GSPP model has also been used to solve classical berth allocation problems

with a fine discretization of the berthing space, and the results show that the model is very effective.

The presented set partitioning models contain many variables and therefore several variable reduction methods are proposed and evaluated. These novel column reduction techniques for the BACAP can reduce the number of columns by up to 90% for some benchmarks. By using the proposed reduction techniques, in most cases, we create models that can be handled in the memory available in current computers, however this is not always the case. We believe that the reduction techniques can be generalized to other variants of the berth allocation problem.

A convenient property of the proposed solution method is that most of the work is done by a black-box IP solver (in this case CPLEX). This means that the solution approach will automatically benefit from new developments in solver techniques and will also benefit from future hardware improvements that are supported by the underlying IP solver (for example a far more massive parallelism).

Future research could be directed towards including more constraints in the model, if that is deemed necessary to apply the model in a particular port or it could be directed at designing improved solution methods. Since a major limitation of the proposed model is the rapid growth in the number of variables with increase in problem size, a natural extension of the current work is to attempt to generate variables dynamically using delayed column generation and solve the model using a branch-and-price algorithm.

## CHAPTER 3

# Improved formulations and an adaptive large neighborhood search heuristic for the integrated berth allocation and quay crane assignment problem

---

This chapter focuses on the integrated berth allocation and QC assignment problem. A number of inequalities to improve state-of-the-art formulations and an adaptive large neighborhood search (ALNS) are presented. The ALNS has various operators that work on time, berth and quay crane assignment levels. Computational results reveal that the valid inequalities and the variable fixing techniques improve many of the best known bounds in the literature, and the ALNS outperforms the state-of-the-art heuristics for many instances.<sup>1</sup>

---

<sup>1</sup>C. Iris, D. Pacino, S. Ropke, "Improved formulations and an adaptive large neighborhood search heuristic for the integrated berth allocation and quay crane assignment problem", 2015, under review

### 3.1 Introduction

Recent statistics show that World container port throughput is increased by an estimated 5.6% to 651.1 million Twenty-foot Equivalent Units (TEUs) in 2013 UNCTAD (2014). The same report illustrates that large container terminals can process more than 30 million containers a year. These high volumes have increased the need for more efficient container terminal operations. Due to the increased number of operations and the interconnection between them, the complexity of planning has also increased tremendously. In this respect, the use of operations research techniques became more popular (see following reviews on container terminal problems and operations research: Steenken et al. (2004), Stahlbock and Voß (2007), Bierwirth and Meisel (2010), Bierwirth and Meisel (2014)).

The productivity of a container terminal heavily relies on the efficient use of its resources. Focusing on quayside operations, Quay Crane (QC) management, and the usage of the berthing area are among some of the most important container terminal planning problems. The optimization of vessels' berthing positions and their respective QC assignments are two problems that are mostly covered as two separate cases in the literature. These problems are however linked. The vessel's handling time primarily depends on the number of containers to be handled and the number of assigned QCs. Berth allocation, which heavily depends on the handling time of the vessel, is thus a function of the QC assignment. Additionally, the QC assignment problem of each vessel requires information about the berthing start and end time which are outcomes of the berth allocation. Due to these bidirectional links, integrating these two problems will result in a better planning of the terminal operations.

In the literature, the integrated problem in which only the number of QCs is a decision variable is called Berth Allocation and quay Crane Assignment Problem; BACAP (Meisel and Bierwirth (2009); Giallombardo et al. (2010), etc.), when determination of the specific QC assignment is involved, the problem is called Berth Allocation and quay Crane Assignment (Specific) Problem; BACASP (Turkogullari et al. (2014); Imai et al. (2008); Liu et al. (2006)).

In this chapter, we consider the BACAP as introduced in Meisel and Bierwirth (2009) where the berthing start/end time, berthing position and the number of QCs to work at each time period are decided for each vessel. It is assumed that the number of QCs can change within vessel's berthing interval (time-variant BACAP). Additionally, it is assumed that there is a decrease in the available QC hours due to interference between QCs. It is also assumed that there will be an increase in the QC hours required to serve a vessel when it is not berthed at its desired berthing position. This increase is due to the rearrangement of

QC operations for that specific vessel and the rehandling of containers which are already positioned according to the desired position.

The BACAP naturally lends itself towards a description in a two dimensional space. One dimension is spatial (i.e. the quay partition) which could be discrete (Cordeau et al. (2005)), continuous (Lee, Chen, and Cao (2010a)), or hybrid. The other dimension is temporal (i.e. the planning horizon) which could be static (Park and Kim (2003)), dynamic (Imai et al. (2008)), or cyclic (Jin, Lee, and Hu (2015), Imai et al. (2014)). Figure 4.1 shows this representation where the vertical axis represents the spatial dimension and the horizontal axis the temporal dimension. In this representation, a vessel is a rectangle whose time dimension is dependent on the number of assigned QCs at each time unit. In this chapter, the spatial dimension is considered to be a continuous berth, where we assume that a vessel can berth at any 10-meter point along the quay (i.e. the quay is discretized for each 10 m). The temporal property is dynamic, because the arrival time of each vessel imposes a bound for the berthing start time, and we assume that all parameters are known in advance (see Golias, Portal, Konur, Kaisar, and Kolomvos (2014) for a berth scheduling problem under uncertainty).

The contribution of this chapter is mainly two-fold. First, we introduce novel valid inequalities and variable fixing methods that improve the state-of-the-art compact model (Meisel and Bierwirth (2009)). Some of these inequalities focus on the vessels' berthing orders, while some others reformulate new bounds on variables. With small changes, these inequalities can also be used for similar problems. Secondly, we present, to the best of the authors knowledge, the first Adaptive Large Neighborhood Search (ALNS) algorithm for the BACAP. ALNS was first proposed by Ropke and Pisinger (2006), (see also Pisinger and Ropke (2007)), and has been used for various problems and has proved its efficiency on the complex optimization problems (see Coelho, Cordeau, and Laporte (2012), Muller, Spoorendonk, and Pisinger (2012), etc.).

Computational results reveal that the performance of the model from Meisel and Bierwirth (2009) has been significantly increased with the addition of our inequalities. The results also show improvements on the bounds presented in Iris et al. (2015b). The performance of each individual family of inequalities is also discussed. The proposed ALNS outperforms the state-of-the-art heuristics of Meisel and Bierwirth (2009), both with respect to the quality of the solutions and the computational performance in many instances. New best upper and lower bounds are found for all instances.

The remainder of this chapter is organized as follows. In Section 6.2, we present a brief literature review for the BACAP and its variants. In Section 3.3, we present a formal definition of the BACAP. The mathematical model proposed by Meisel and Bierwirth (2009), valid inequalities and variable fixing methods



for this formulation are described in Section 6.4. In Section 3.5, we present an ALNS for the BACAP. Computational results are discussed in Section 3.6. Finally, we present conclusions and future research directions in the last section.

## 3.2 Literature Review

The BACAP has attracted many researchers in the field. Extensive reviews on BACAP and BACASP literature can be found in Bierwirth and Meisel (2010), Bierwirth and Meisel (2014), Carlo, Vis, and Roodbergen (2013), Iris et al. (2015b) in which the authors cluster papers according to problem structure, objective function characteristics and solution methods. One of the first works on BACAP is presented in Park and Kim (2003). The authors solve the BACAP in a first stage with a subgradient optimization strategy. The results are then used in a second stage to solve the BACASP with dynamic programming. Meisel and Bierwirth (2009) present a different BACAP model. The authors propose various heuristics (tabu search (TS), first-come-first-served (FCFS) heuristics, and squeaky wheel optimization (SWO)) to solve this problem.

Another paper, presented by Liang et al. (2009), determines the berthing positions, berthing start/end times and number of QCs to serve each vessel. They assume a discrete berth allocation (i.e. vessels fit into discrete berths) and a time-invariant QC allocation policy (i.e. the number of QC operating on a vessel does not change in time). They solve the problem with a hybrid genetic algorithm. Giallombardo et al. (2010) have focused on a variant of BACAP called the tactical berth allocation problem (TBAP). This problem uses the concept of QC profiles. A QC profile includes the number of QCs assigned to a vessel along the time steps that the vessel is berthed at the port. This profile also holds some real-world requirements imposed by some terminals such as; QC movements only at shift changes, vessel priorities in terms of the number of QCs, etc. The authors solve the problem via a two-stage heuristic algorithm which combines tabu search and dual bound properties. Vacca et al. (2013) focus on the formulation of Giallombardo et al. (2010) and they propose the first exact decomposition framework for this variant of the BACAP. Lalla-Ruiz, González-Velarde, Melián-Batista, and Moreno-Vega (2014) also focus on this TBAP formulation, and implement a genetic algorithm to solve the problem. Finally, Iris et al. (2015b) formulate various set partitioning formulations and column reduction techniques for the BACAP and its variants (time-variant/time-invariant QC allocation, finer/hybrid berth discretization, etc.). The authors improve almost all known bounds for the instances presented by Meisel and Bierwirth (2009).

Liu et al. (2006) is one of the first papers on BACASP. The authors assume that

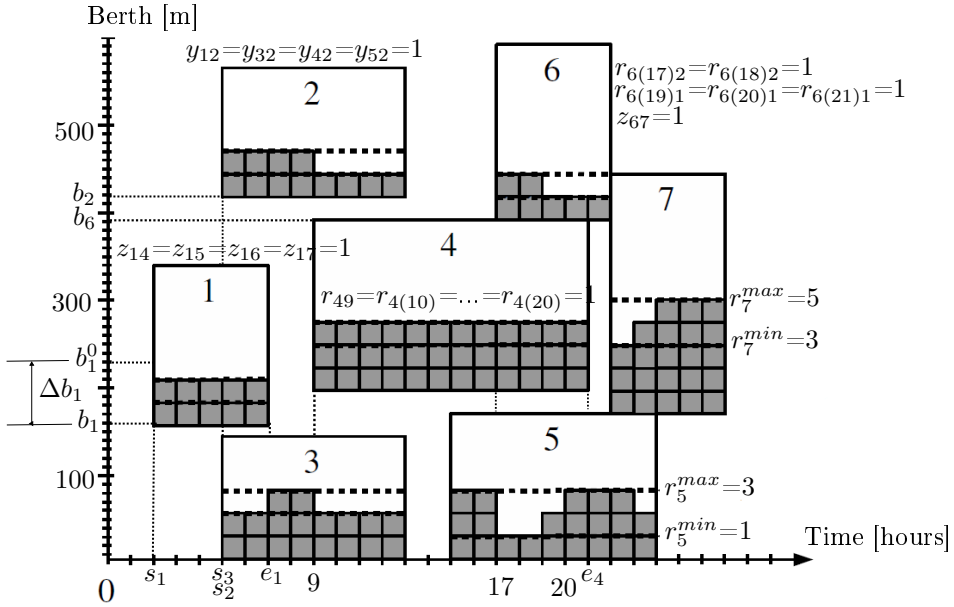
the berthing position for each vessel is known in advance (i.e. one important component of BAP is given as a parameter) and they determine vessel berthing start/end times, the number of QCs to assign to each vessel and which specific QC would be assigned. They propose two greedy heuristics to solve this problem. Imai et al. (2008) presents a novel mathematical model which formulates a complete BACASP. They consider a discrete berth partition where every vessel can fit in exactly one berth. And they do not consider the relationship between the handling time and the number of cranes assigned to a vessel. The authors propose a genetic algorithm to solve the problem. Ursavas (2014) also focus on a BACASP with discrete berths. The author considers time-variant QC allocation, proposes a bi-objective mathematical model and implements a decision support system. Turkogullari et al. (2014) also formulated a model for the BACASP. The authors emphasize that for large scale instances the model is not effective. Hence, they propose a post-processing cutting plane algorithm over the results of a BACAP solution. Experiments show that the largest instances can be solved to optimality with this method. Their main assumption lays in the fact that the number of QCs assigned to a vessel cannot be changed over time (time-invariant BACAP, see also Iris et al. (2015b)). Rodriguez-Molins et al. (2014b) have focused on a BACASP with both time-variant/invariant QC allocation. The time-variant QC version is designed to assign the QCs for specific holds. The authors have proposed a Greedy Randomized Adaptive Search Procedure (GRASP) heuristic and showed that the algorithm outperforms traditional heuristics like FCFS, etc. Recently, Li, Sheu, and Gao (2015a) have focused on a BACASP in which QC coverage ranges are also considered. The authors present a novel mathematical model which has many BACAP considerations from Meisel and Bierwirth (2009) model. They also propose a heuristic algorithm based on spatio-temporal conflicts analysis.

### 3.3 Problem Description

The BACAP studies in this chapter aims at finding a berthing start time and position for each vessel in the planning horizon. Moreover, the berthing end time is calculated as a function of the number of assigned QCs in each time period. We consider a continuous berth and discretize time in units on one hour.

Figure 4.1 shows an example BACAP solution in a time/quay diagram. In this example, seven vessels are berthed. Each vessel is represented by a rectangle showing the time and space occupied by the vessel. The smaller rectangles in gray indicate the vessels' QC assignments, each representing one QC. Every berthed vessel has an upper and lower limit on the number of assignable QCs

$(r_i^{min}, r_i^{max})$ . These values are determined by contracts between vessel operators and the terminal, and by the size of the ship. A limited number of QCs are available in the berth and this determines the maximum number of QCs that can be assigned at any point in time. Every vessel has also a berthing start ( $s_i$ ) and end time ( $e_i$ ) on the horizontal axis. The vertical axis includes the berthing position of each vessel ( $b_i$ ), and each vessel has a desired berthing position parameter ( $b_i^0$ ). The deviation from desired berthing position is  $\Delta b_i$ . There are also variables regarding the QC assignment ( $r_{itq}$ ) and vessel ordering ( $z_{ij}$ ). The example is further explained in the next section.



**Figure 3.1:** Example of BACAP Meisel and Bierwirth (2009)

There are two considerations about the utilization of resources in the terminal in Meisel and Bierwirth (2009). The authors argue that the assigned number of QC for each vessel cannot be completely used due to the interference between QCs. This means when  $q$  QCs are assigned to a vessel, the productivity is  $q^\alpha$  QC-hours, where  $\alpha$  is an exponent of interference ( $0 \leq \alpha \leq 1$ ). Another aspect which is considered is the increase in the horizontal transportation when the vessel is not berthed at its desired berthing position. This means that there is an increase in the QC hours needed to carry-out the work (i.e. fulfill the QC capacity requirement) on a vessel. This increase is proportional to the distance of the berthing position from its desired position.

Meisel and Bierwirth (2009) consider a combination of time-dependent costs and QC assignment costs for the objective function. The time-dependent costs are entailed to the berthing start and end time, while QC dependent cost is a function of how many QCs are assigned to a given vessel. The model includes three time-dependent cost components: speedup cost, delay cost and penalty cost. It is assumed that vessels can speedup in the open sea and berth earlier than the Expected Time of Arrival (ETA). For each time unit earlier than ETA, a speedup cost must be paid. It is, however, assumed that a fixed Earliest Start Time (EST) for each vessel exists. The problem does not allow any berthing start time before EST, since it is not feasible with respect to the maximum speed of a vessel. The problem also penalizes lateness for each vessel. If a vessel finishes its operations and departs (berthing end time) after Expected Finishing Time (EFT), there is a cost of lateness for each late time period. Moreover, if the berthing end time is later than the Latest Finishing Time (LFT), a one-time penalty cost is added to the cost function. There exists a trade-off between accelerating the handling time (which is also a function of the berthing position) of a vessel and the corresponding number of QCs assigned to the vessel. Finally, the cost of operating all assigned QCs of a vessel is also added to model.

### **3.4 Mathematical Model: Meisel and Bierwirth (2009) Model**

The list of notations, i.e. parameters, decision variables used in the model, are listed in Table 6.1.

**Table 3.1:** BACAP mathematical notation

Parameters and sets:	
$V$	Set of vessels to be served, $V \in \{1, 2, \dots, N\}$ , where $N$ is the number of vessels to berth
$T$	Set of time periods (1 hour), $T \in \{0, 1, \dots, H - 1\}$ , where $H$ is the planning horizon
$L$	Length of the quay given in 10s of meters
$l_i$	Length of vessel $i \in V$ in 10s of meters
$b_i^0$	Desired berthing position of vessel $i \in V$
$m_i$	Quay crane capacity demand of vessel $i \in V$ (i.e. total QC-hours needed)
$r_i^{min}$	Minimum number of QCs agreed to serve vessel $i \in V$ simultaneously
$r_i^{max}$	Maximum number of QCs agreed to serve vessel $i \in V$ simultaneously
$R_i$	Set of number of QCs assignable to vessel $i \in V$ , where $R_i \in \{r_i^{min}, \dots, r_i^{max}\}$
$ETA_i$	Expected time of arrival of vessel $i \in V$
$EST_i$	Earliest starting time of vessel $i \in V$
$EFT_i$	Expected finishing time of vessel $i \in V$
$LFT_i$	Latest finishing time of vessel $i \in V$
$c_i^1$	Speedup cost of vessel $i \in V$ on its journey to catch a berthing time earlier than $ETA_i$
$c_i^2$	Cost of exceeding the expected finishing time $EFT_i$ for vessel $i \in V$
$c_i^3$	Penalty cost by exceeding $LFT_i$ for vessel $i \in V$
$c^4$	Cost rate per QC-hour of operations
$\alpha$	Interference exponent for the QCs. Only $q^\alpha$ effective QC hours are obtained when assigning $q$ QCs to a ship for one hour
$\beta$	Coefficient of increase in the QC capacity demand with deviation from desired berthing position. A vessel $i$ placed at position $b_i$ needs $(1 +  b_i^0 - b_i \beta)m_i$ effective QC hours
$M$	A large positive number
$Q$	Available number of QCs
Decision variables:	
$b_i \in \mathbb{Z}^+$	Berthing position of vessel $i \in V$
$s_i \in \mathbb{Z}^+$	Berthing start time of vessel $i \in V$
$e_i \in \mathbb{Z}^+$	Berthing end time (time when the handling ends) of vessel $i \in V$
$r_{it} \in \mathbb{B}$	1; if vessel $i \in V$ is at berth being processed in period $t$ , 0 otherwise
$r_{itq} \in \mathbb{B}$	1; if there is exactly $q$ QCs assigned to vessel $i$ in period $t$ , 0 otherwise
$\Delta b_i \in \mathbb{Z}^+$	Deviation from desired berthing position when vessel $i$ is in position $b_i$ , $\Delta b_i =  b_i^0 - b_i $
$\Delta ETA_i \in \mathbb{Z}^+$	Earliness of vessel $i$ to reach start-time $s_i$ , where $\Delta ETA_i =  ETA_i - s_i $
$\Delta EFT_i \in \mathbb{Z}^+$	Tardiness of vessel $i \in V$ when operations are finished later than expected finishing time, $\Delta EFT_i =  e_i - EFT_i $
$u_i \in \mathbb{B}$	1; if finishing time of vessel $i \in V$ exceed latest finishing time, 0 otherwise
$y_{ij} \in \mathbb{B}$	1; if vessel $i \in V$ is berthed below vessel $j \in V$ in berth area, i.e. $b_i + l_i \leq b_j$ , 0 otherwise
$z_{ij} \in \mathbb{B}$	1; if handling of vessel $i \in V$ ends no later than handling of vessel $j \in V$ starts in berth area, 0 otherwise

Let us now introduce the mathematical model:

$$\min \sum_{i \in V} (c_i^1 \Delta ETA_i + c_i^2 \Delta EFT_i + c_i^3 u_i + c^4 \sum_{t \in T} \sum_{q \in R_i} r_{itq} q) \quad (3.1)$$

subject to

$$\sum_{t \in T} \sum_{q \in R_i} q^\alpha r_{itq} \geq (1 + \Delta b_i \beta) m_i \quad \forall i \in V \quad (3.2)$$

$$\sum_{i \in V} \sum_{q \in R_i} q r_{itq} \leq Q \quad \forall t \in T \quad (3.3)$$

$$\sum_{q \in R_i} r_{itq} = r_{it} \quad \forall i \in V, \forall t \in T \quad (3.4)$$

$$\sum_{t \in T} r_{it} = e_i - s_i \quad \forall i \in V \quad (3.5)$$

$$(t + 1) r_{it} \leq e_i \quad \forall i \in V, \forall t \in T \quad (3.6)$$

$$r_{it} t + H(1 - r_{it}) \geq s_i \quad \forall i \in V, \forall t \in T \quad (3.7)$$

$$\Delta b_i \geq b_i - b_i^0 \quad \forall i \in V \quad (3.8)$$

$$\Delta b_i \geq b_i^0 - b_i \quad \forall i \in V \quad (3.9)$$

$$\Delta ETA_i \geq ETA_i - s_i \quad \forall i \in V \quad (3.10)$$

$$\Delta EFT_i \geq e_i - EFT_i \quad \forall i \in V \quad (3.11)$$

$$Mu_i \geq e_i - LFT_i \quad \forall i \in V \quad (3.12)$$

$$b_j + M(1 - y_{ij}) \geq b_i + l_i \quad \forall i, j \in V, \quad i \neq j \quad (3.13)$$

$$s_j + M(1 - z_{ij}) \geq e_i \quad \forall i, j \in V, \quad i \neq j \quad (3.14)$$

$$y_{ij} + y_{ji} + z_{ij} + z_{ji} \geq 1 \quad \forall i, j \in V, \quad i \neq j \quad (3.15)$$

$$s_i, e_i \in \{EST_i, \dots, H\} \quad \forall i \in V \quad (3.16)$$

$$b_i \in \{0, 1, \dots, L - l_i\} \quad \forall i \in V \quad (3.17)$$

$$\Delta ETA_i, \Delta EFT_i \geq 0 \quad \forall i \in V \quad (3.18)$$

$$r_{itq}, r_{it}, u_i, y_{ij}, z_{ij} \in \{0, 1\} \quad \forall i, j \in V, \forall t \in T, \forall q \in R_i \quad i \neq j \quad (3.19)$$

The objective (6.1) is the minimization of the overall cost of operations. It includes speeding costs (proportional to the number of time periods for which a vessel is early), tardiness cost (proportional to the number of time periods a vessel departs later than EFT), a one time penalty cost for finishing later than LFT, and costs related to QC assignments. The QC assignment cost is a function of the number of QCs and the cost rate per QC-hour. Constraint (3.2) guarantees that the number of efficient QC hours assigned to a vessel meets the

required QC capacity when the deviation factor is taken into account. Constraint (3.3) ensures that the number of QCs used in each period cannot exceed the available number of QCs. Constraint (3.4) guarantees that when a vessel is at berth it will be served by a number of QCs. This constraint ensures the assumption that there are no idle periods when the vessel is at port. Constraint (3.5) represents that the port-stay of a vessel (the right hand side) is equal to the number of periods that the vessel is at berth. Constraint (3.6) and (3.7) set the berthing start and end time. The berthing end time is limited by the planning horizon. The deviation from the expected berthing position is calculated by Constraint (3.8) and (3.9). Constraints (3.10)-(3.12) determine the required speedup time needed to reach  $s_i$ , the tardiness of the operations, and whether a penalty must be paid due to vessel delays. Constraint (3.13) and (3.14) help to link the variables  $y_{ij}$  and  $z_{ij}$ . Constraint (3.13) ensures that if  $y_{ij}$  has a value of one, the berthing position (which corresponds to the fore of vessel) of vessel  $j \in V$  is larger than aft of vessel  $i \in V$ . Constraint (3.14) links the vessels if they follow a predecessors relationship in the time frame. If  $z_{ij}$  has a value of one, the berthing start time for vessel  $j \in V$  is later than the berthing end time for vessel  $i \in V$ . Constraint (3.15) ensures either one vessel should be positioned before or after the other at the berth, if not, the berthing intervals should not overlap. Domains of berthing start and end time variables are illustrated in (3.16). Finally, Constraints (3.18) and (4.7) set the integer and binary properties of the respective decision variables. A constraint that assigns at most one QC-assignment plan for each time unit and for each vessel can be added ( $\sum_{q \in R_i} r_{itq} \leq 1 \quad \forall i \in V, \forall t \in T$ ). The validity of this constraint is evident since it makes sure that there is at most one kind of QC assignment plan for a certain vessel (i.e. one of  $r_i^{min}, r_i^{min} + 1, \dots, r_i^{max} - 1, r_i^{max}$ ) for each period when the vessel is at berth. This constraint is dominated by Constraint (3.4), so the LP relaxation will not be improved, but CPLEX can generate useful cuts out of this constraint.

### 3.4.1 Valid Inequalities and Variable Fixing Methods for BACAP

In order to make the mathematical model more efficient, we propose a novel set of valid inequalities, Lower Bounds (LBs) and variable fixing methods. In the remainder of this chapter, we refer to this enhanced model as BACAP+.

Before describing the inequalities in details, let us formalize the minimum and maximum processing time of a vessel. The minimum processing time can be obtained when there is no deviation from the desired berthing position, and the maximum number of QCs is assigned to the vessel for every time unit during its berthing interval. In (6.19),  $\delta_{min}^i$  is the minimum processing time required to

fulfill the QC capacity demand of vessel  $i$ . In (3.21), the maximum processing time,  $\delta_{max}^i$ , is composed of the maximum possible deviation from desired position, and the minimum number of QC assignments for that vessel. Since we do not allow preemption in operations, the use of  $r_i^{min}, r_i^{max}$  in the denominators is reasonable.

$$\delta_{min}^i = \left\lceil \frac{m_i}{(r_i^{max})^\alpha} \right\rceil \quad \forall i \in V \quad (3.20)$$

$$\delta_{max}^i = \left\lceil \frac{m_i(1 + \beta \max(L - b_0^i, b_0^i))}{(r_i^{min})^\alpha} \right\rceil \quad \forall i \in V \quad (3.21)$$

#### 3.4.1.1 Bounds on $s_i$

We formulate a class of valid inequalities that aim at improving the lower bound of the  $s_i$  (berthing start time) variables. If vessel  $i$  is berthed earlier than vessel  $j$ , the berthing start time of vessel  $j$  ( $s_j$ ) should be equal-to or greater-than the minimum ending time of vessel  $i$  ( $e_i$ ). Otherwise, the *EST* still limits the starting time of vessel  $j$ .

$$(EST_i + \delta_{min}^i)z_{ij} + EST_j(1 - z_{ij}) \leq s_j \quad \forall i, j \in V, \quad i \neq j \quad (3.22)$$

#### 3.4.1.2 Bounds on processing time

The bounds on the processing time can also be tightened by Constraint (3.23) by using the precalculated  $\delta_{min}^i$  and  $\delta_{max}^i$  values.

$$\delta_{min}^i \leq \sum_{t \in T} r_{it} \leq \delta_{max}^i \quad \forall i \in V \quad (3.23)$$

#### 3.4.1.3 Decomposition of $s_i$ and $e_i$

In this family of valid inequalities, we improve the link between the QC assignment and the berthing start/end time variables. We observe that reformulating berthing start and end times variables for each time period can help to obtain



a better link with the QC assignment variables. We propose a new intermediate binary decision variable denoted  $Start_{it}$ . If operations on vessel  $i$  starts at period  $t$ , the variable is one, and it is zero otherwise. We propose seven inequalities ((3.24)-(3.30)) to improve the formulation. First, we guarantee that there can only be one berthing start period for each vessel (Constraint (3.24)), and this period cannot be before EST (Constraint (3.25)). Then, the link between  $Start_{it}$  and  $s_i$  is formulated in Constraints (3.26)-(3.27). Constraint (3.28) guarantees that if a period is a starting period, then the previous period cannot hold any QC assignment. We can also formulate an alternative version of Constraint (3.28). If a given period is a berthing start period, all previous periods should hold no QC assignment. If that period is not a starting period, no information can be derived for previous periods. Constraint (3.29) presents this link. Finally, Constraint (3.30) guarantees that if there is a period in which there is a QC assignment for a given vessel ( $r_{it} = 1$ ), then either it is the starting period ( $Start_{it} = 1$ ) or the QC operations have already been continuing from the previous period ( $r_{it-1} = 1$ ).

$$\sum_{t \in T} Start_{it} = 1 \quad \forall i \in V \quad (3.24)$$

$$Start_{it} = 0 \quad \forall i \in V, \forall t \in T : \{t < EST_i\} \quad (3.25)$$

$$\sum_{t \in T} t Start_{it} = s_i \quad \forall i \in V \quad (3.26)$$

$$r_{i0} = Start_{i0} \quad \forall i \in V \quad (3.27)$$

$$r_{it-1} + Start_{it} \leq 1 \quad \forall i \in V, \forall t \in T : \{t > EST_i\} \quad (3.28)$$

$$\sum_{t' \in \{EST_i, \dots, t-1\}} r_{it'} + (Start_{it} - 1)H \leq 0 \quad \forall i \in V, \forall t \in T : \{t > EST_i\} \quad (3.29)$$

$$r_{it} \leq r_{it-1} + Start_{it} \quad \forall i \in V, \forall t \in T : \{t \geq EST_i\} \quad (3.30)$$

We also use a symmetrical formulation for the end of operations, where  $End_{it} = \{0, 1\}$  is the auxiliary binary variable which is one if operations of vessel  $i$  end at period  $t$ , and zero otherwise. The difference from the previous inequalities is in the linking constraints. In Constraint (3.37), we guarantee that if a QC assignment exists in one period for a vessel ( $r_{it-1} = 1$ ), then either the next period is the ending period ( $End_{it} = 1$ ), or there are still QC operations going

on the vessel for the next period ( $r_{it} = 1$ ). Constraints (3.31)-(3.36) can be interpreted in the same way as Constraints (3.24)-(3.29).

$$\sum_{t \in T \cup \{H\}} t \text{End}_{it} = e_i \quad \forall i \in V \quad (3.31)$$

$$\sum_{t \in T \cup \{H\}} \text{End}_{it} = 1 \quad \forall i \in V \quad (3.32)$$

$$\text{End}_{it} = 0 \quad \forall i \in V, \forall t \in T \cup \{H\} : \{t < \text{EST}_i + \delta_{min}^i\} \quad (3.33)$$

$$r_{iH-1} = \text{End}_{iH} \quad \forall i \in V \quad (3.34)$$

$$r_{it} + \text{End}_{it} \leq 1 \quad \forall i \in V, \forall t \in T \cup \{H\} : \{t > 0\} \quad (3.35)$$

$$\sum_{t' \in \{t, \dots, H\}} r_{it'} + (\text{End}_{it} - 1)H \leq 0 \quad \forall i \in V, \forall t \in T \cup \{H\} : \{t > \text{EST}_i\} \quad (3.36)$$

$$r_{it-1} \leq r_{it} + \text{End}_{it} \quad \forall i \in V, \forall t \in T \cup \{H\} : \{t > 0\} \quad (3.37)$$

#### 3.4.1.4 Tightening the Big-M values

Constraint (3.7) in the original formulation can be tightened by reformulating *Big - M* which is  $H$ . We know that it cannot be higher than  $H - \delta_{min}^i$ . We, thus, replace Constraint (3.7) with Constraint (3.38) in the BACAP model.

$$r_{it}t + (H - \delta_{min}^i)(1 - r_{it}) \geq s_i \quad \forall i \in V, \forall t \in T \quad (3.38)$$

#### 3.4.1.5 Set-partitioning inequalities

Iris et al. (2015b) formulate a generalized set partitioning problem (GSPP) formulation for the BACAP. The authors also propose two preprocessing (or column reduction methods) to reduce the number of columns. Each column/variable in the formulation by Iris et al. (2015b) corresponds to an assignment of a vessel to a berthing position and a berthing start and end time. After column reduction, we can go through all columns for a single vessel ( $i$ ) and record earliest and latest berthing start ( $\text{Min}S_i$ ,  $\text{Max}S_i$ ) and end time ( $\text{Min}E_i$ ,  $\text{Max}E_i$ ).

Similarly, we can collect minimum and maximum possible berthing positions ( $MinB_i, MaxB_i$ ). With these parameters, we bound the decision variables in Meisel and Bierwirth (2009). Technically speaking, these constraints are not valid inequalities since they can cut away feasible integer solutions. This is because the column reduction methods in Iris et al. (2015b) were based on objective value considerations. However, the constraints do not cut away the optimal solution and therefore they do not alter the exactness of the model. The collected data provide us better bounds on  $s_i, e_i$  and it allows us to eliminate many of the  $r_{it}$  and  $r_{itq}$  variables. Then we can write following inequalities:

$$MinS_i \leq s_i \leq MaxS_i \quad \forall i \in V \quad (3.39)$$

$$MinE_i \leq e_i \leq MaxE_i \quad \forall i \in V \quad (3.40)$$

$$MinB_i \leq b_i \leq MaxB_i \quad \forall i \in V \quad (3.41)$$

$$r_{it} = 0 \quad \forall i \in V, \forall t \in T : \{t > MaxE_i \parallel t < MinS_i\} \quad (3.42)$$

Constraints (3.39), (3.40), (3.41) bound the berthing start time, end time and berthing position of vessel  $i$ , respectively. Constraint (3.42) can eliminate QC assignment variables when period  $t$  is greater than maximum feasible berthing end period or smaller than minimum feasible berthing start period for vessel  $i$ .

#### 3.4.1.6 Fixing $z_{ij}$ and $y_{ij}$ variables

By considering minimum and maximum processing times, and the bounds obtained in Section 3.4.1.5, we can fix and tighten some of the  $z_{ij}, y_{ij}$  variables.

$$z_{ij} = 1 \quad \forall i, j \in V : \{i \neq j \wedge MinS_j \geq MaxE_i\} \quad (3.43)$$

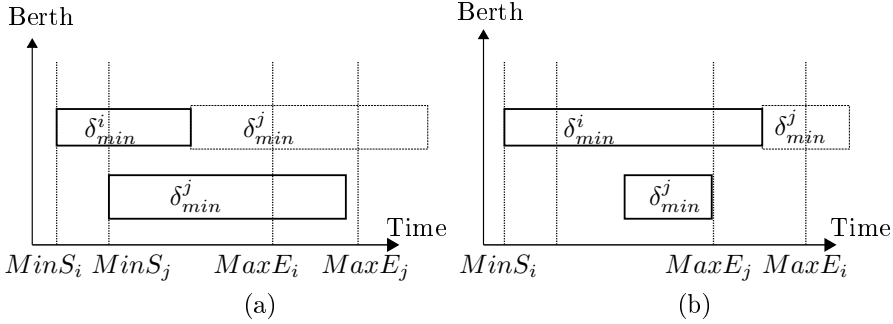
$$y_{ij} = 1 \quad \forall i, j \in V : \{i \neq j \wedge MaxB_i + l_i \leq MinB_j\} \quad (3.44)$$

Constraint (3.43) shows that if the minimum berthing start time of vessel  $i$  is greater-than or equal-to the maximum berthing end time of vessel  $j$ , then vessel  $i$  is always berthed earlier than vessel  $j$  ( $z_{ij}=1$ ). Constraint (3.44) works in the same way for the  $y_{ij}$  variables.

We can also state that vessel  $j$  cannot be berthed later than vessel  $i$  if and only if the earliest known berthing start time of vessel  $i$  ( $MinS_i$ ) added to the minimum processing times of vessel  $i$  and  $j$  is larger than the known maximum berthing end time of a vessel  $j$ . This means that these two vessels should either overlap on time or vessel  $j$  should end its berthing earlier than the berthing start time of vessel  $i$ . Figure 3.2 illustrates two alternatives that include two vessels which hold the aforementioned condition, and these vessels must be at berth at the same time. This relation can be formulated with Constraint (3.45).

$$z_{ij} = 0 \quad \forall i, j \in V : \{i \neq j \wedge MinS_i + \delta_{min}^i + \delta_{min}^j \geq MaxE_j\} \quad (3.45)$$

Constraints similar to (3.45) can be formulated for the  $y_{ij}$  variables, but the bounds obtained for minimum and maximum possible berthing position are not strong enough to justify their use.



**Figure 3.2:** Different scheduling alternatives for Constraint (3.45)

#### 3.4.1.7 Bounds on $\Delta b_i$

The last two inequalities focus on the objective function. We should note that these constraints are not valid inequalities, but adding them to the model does not invalidate the exactness of the model. Let us assume that an upper bound of  $\bar{z}$  is known for this problem. We can calculate the minimum number of QC hours needed to serve all vessels,  $\theta = \sum_{i \in V} r_i^{min} \left\lceil \frac{m_i}{(r_i^{min})^\alpha} \right\rceil$ . We can extract a LB on the overall cost as shown in Iris et al. (2015b) (See B.2 for the complete algorithm to obtain the LB). This lower bound on the objective is called  $\underline{z}^1$  and zero deviation is assumed for all vessels in order to calculate the  $\underline{z}^1$  (i.e.  $\Delta b_i = 0 \quad \forall i \in V$  in  $\underline{z}^1$ ).

We adjust  $\underline{z}^1$  by adding the approximated QC operations cost with respect to the deviation from the desired position ( $\sum_{i \in V} (c_4(1 + \beta \Delta b_i) m_i)$ ). In order to not include QC costs twice, we subtract the minimum cost of QC operations obtained by  $c_4 \theta$ . It follows that the resulting approximation of the objective function must be less or equal-to the upper bound value, and this forms Constraint (3.46).

$$\underline{z}^1 + c_4 \left( \sum_{i \in V} (1 + \beta \Delta b_i) m_i - \theta \right) \leq \bar{z} \quad (3.46)$$

The term  $(1 + \beta \Delta b_i) m_i$  is the number of "pure" QC hours that we need to spend on vessel  $i$  given a deviation of  $\Delta b_i$  from the desired position. However, if we are using more than one QC per hour the actual number of QC hours needed increases because of the interference exponent  $\alpha$ . The actual number of QCs used per hour will be determined by the model, but it can never be less than  $r_i^{min}$ . Assuming that we use the minimum number of QCs we can find for how many hours ( $x$ ) we need to operate  $r_i^{min}$  QCs in order to finish the task by solving following equation.

$$x (r_i^{min})^\alpha = (1 + \beta \Delta b_i) m_i \Rightarrow x = (1 + \beta \Delta b_i) m_i / (r_i^{min})^\alpha$$

In order to know how many QC hours we at least need to spend on vessel  $i$  given a deviation of  $\Delta b_i$  we therefore compute

$$((1 + \beta \Delta b_i) m_i / (r_i^{min})^\alpha) r_{min}$$

since  $x$  measured for how many hours we need to use  $r_{min}$  QCs. This means that we can improve the inequality to (3.47).

$$\underline{z}^1 + c_4 \left( \sum_{i \in V} \left( (1 + \beta \Delta b_i) m_i \frac{r_i^{min}}{(r_i^{min})^\alpha} \right) - \theta \right) \leq \bar{z} \quad (3.47)$$

The (3.47) is a strengthening of (3.46) if  $\alpha < 1$  and  $r_i^{min} > 1$  for all  $i \in V$ . The only decision variable in Constraint (3.47) is  $\Delta b_i$  and this constraint will be used in BACAP+.

#### 3.4.1.8 Bound on time-dependent objective part

Finally, we know that the minimum cost of QC operations is  $c_4\theta$ , we can set a bound on the objective function for the remaining cost components which are dependent on the vessel berthing start and end times. The sum of speedup, lateness and penalty costs should be less than upper bound on overall objective subtracted by LB from QC assignment costs. This is achieved with Constraint (3.48).

$$\sum_{i \in V} (c_i^1 \Delta ETA_i + c_i^2 \Delta EFT_i + c_i^3 u_i) + c_4 \theta \leq \bar{z} \quad (3.48)$$

### 3.5 Adaptive Large Neighborhood Search (ALNS) heuristic for BACAP

ALNS (Ropke and Pisinger (2006)) extends the Large Neighborhood Search (LNS) of Shaw (1998). ALNS is a search algorithm based on a destruction and construction principle. Once an initial solution is found, part of the candidate solution is destroyed by an operator, while keeping the remaining part fixed. A new solution is then found by repairing the destroyed part with a repair operator. These two steps are iterated until some termination criterion. The main difference between ALNS and LNS is that ALNS has multiple destroy/repair operators and the selection of operators is dynamically managed as the search progresses.

In order to efficiently search the solution space, we generate all candidate assignments for each vessel a priori and apply our ALNS on top of all candidate assignments of all vessels. Each assignment holds a feasible solution of a single vessel. For each vessel  $i \in V$ , each assignment  $j$  holds a cost  $c_j$ , a berthing position, berthing start and end time, and a QC plan. The QC plan includes the information on how many time units each number of QCs will be used in that assignment. Iris et al. (2015b) have proved that there are always two QC numbers which can form a QC plan with minimum cost, these two values are stored in the assignment. The assignment does not give any information about the specific time in which these number of QCs will be used. For example, a QC plan might hold that the assignment will use 3 QCs for 4 time units and 4 QCs for 5 time units. The cost of an assignment includes the time-dependent cost components (speedup, lateness, penalty costs), and the cost of QC plan. Since we generate all related information about berthing start/end times and berthing

position a priori, the time-dependent cost component of an assignment can be calculated easily. The cost of QC plan can be calculated with the information which is held in the assignment (recalling above example;  $(3 \cdot 4 + 4 \cdot 5) \cdot c^4$ ).

A candidate solution for the BACAP ( $X$ ) is a set of assignments that satisfies all the constraints presented in Section (6.4). A candidate solution also contains a detailed QC assignment plan which is the number of QCs that will work on each vessel in each period. An optimal solution is a candidate solution with minimum  $Z(X)$  where  $Z(X)$  accounts for objective function value of candidate solution  $X$ . The set of assignments of vessel  $i$  will be denoted by  $\Omega_i$ , while the set of all assignments is denoted by  $\Omega$ .

The outline of the BACAP ALNS is presented in Algorithm 2. The search requires an initial solution which could be generated by a heuristic or a mathematical model. In this paper, we obtain an initial solution with a construction heuristic. The ALNS attempts to improve this initial solution by removing a subset of assignments from the solution (Note that there is only one assignment of each vessel in a solution) and by inserting new assignments of removed vessels back into the solution sequentially. Since the problem is a minimization problem, if the new solution has an objective value lower than current best solution, it is labeled as the best solution (line 7). Whenever all vessels which were removed are reinserted in the current solution, there is an acceptance criterion (line 9) which decides whether the new assignment will be accepted. This procedure continues until a certain termination criterion is met. In order to promote well-performing operators, the weight (i.e. selection probabilities consequently) of each operator is updated periodically (line 11).

The ALNS has been applied successfully to many optimization problems. In the area of container terminal optimization, Cordeau, Laporte, Moccia, and Sorrentino (2011) have proposed an ALNS for yard assignment problem in a car transshipment terminal. The problem studied in Cordeau et al. (2011) has properties that are related to the BAP, hence our approach uses some similar operators (see also Hansen, Oguz, and Mladenovic (2008)). The authors point out the efficiency of the algorithm on real-life instances. Gharehgozli, Laporte, Yu, and de Koster (2015) have proposed an ALNS for the twin crane scheduling problem in container terminals. The authors have showed the efficiency of the

algorithm on a large set of instances.

---

**Algorithm 2:** BACAP ALNS

---

**Input** : An initial solution  $X$ ,  $\phi \in \{1, \dots, N\}$ , initial operator weights

```

1  $X_{best} \leftarrow X$ 
2 while termination criteria is not met do
3    $\bar{X} \leftarrow X$ 
4   Select and apply one operator to remove assignments of  $\phi$  vessels from  $\bar{X}$ 
5   Select and apply one operator to reinsert assignments of  $\phi$  removed vessels back into  $\bar{X}$ 
6   if  $Z(\bar{X}) < Z(X_{best})$  then
7      $X \leftarrow \bar{X}$ ,  $X_{best} \leftarrow \bar{X}$ 
8   else
9     if  $\bar{X}$  satisfies the acceptance criterion then
10       $X \leftarrow \bar{X}$ 
11   adjust operator weights
12 return  $X_{best}$ 

```

---

The main components of BACAP ALNS are operators, adaptive search engine, adaptive weight adjustments, acceptance and termination criterion.

- Operators: To guarantee a diversified search, we propose four destroy and two insertion operators. Details of operators will follow in this section.
- Adaptive Weight Adjustment: Each operator  $i$  is assigned a weight ( $w_i$ ) and a score ( $\pi_i$ ). After each  $\delta$  iterations, all weights are recalculated. The weight of each operator is updated by considering the score of operator in the last  $\delta$  iterations and the current weight value. Initially, all weights are equal to one. While, the scores of all operators are updated after each iteration, and every  $\delta$  iterations they are all set to zero again. The score of each operator is updated as follows:

$$\pi_i = \begin{cases} \pi_i + \sigma_1 & \text{if Condition 1} \\ \pi_i + \sigma_2 & \text{if Condition 2} \\ \pi_i + \sigma_3 & \text{if Condition 3} \\ \pi_i & \text{if Condition 4} \end{cases}$$

where Condition 1: operator  $i$  obtains a new best solution of all iterations, Condition 2: operator  $i$  results in a solution which is improving the current objective function, Condition 3: operator  $i$  does not yield a better



objective value, however the acceptance criterion accepts it, Condition 4: the acceptance criterion rejects the solution. The higher operator performance results in a better score. At end of each iteration, we update the scores of both destroy and repair operators. After scores have been summed up for  $\delta$  iterations, the weights can be updated. The weight of operator  $i$  is updated in following way:

$$w_i = \begin{cases} w_i & \text{if } \Psi_i = 0 \\ (1 - \eta)w_i + \frac{\eta\pi_i}{\Psi_i} & \text{if } \Psi_i \neq 0 \end{cases}$$

Let us assume that  $w_i$  is the weight of operator  $i$ , and  $\Psi_i$  is the number of times that the operator is used for the last  $\delta$  iterations. If the operator is not used, the weight is kept the same, otherwise it is updated. Finally,  $\eta \in [0, 1]$  is the reaction factor which reflects the balance between previous weight and its updated value (Ropke and Pisinger (2006)). A low reaction factor makes the weights evolve in a slow and steady fashion.

- **Adaptive Search Engine:** The selection of which operator to apply is managed by a roulette-wheel technique. The weight of each operator is used to obtain the selection probability of that operator. If there are  $m$  operators, operator  $i$  would have a selection probability of  $w_i / \sum_{j=1}^m w_j$ . These probability values are used to generate cumulative probability distribution (CDF) with all operators. A uniform random number from the range  $[0,1]$  is drawn and the inverse of the CDF for that number points out which operator should be selected.
- **Accepting Criteria:** Simulated Annealing (SA) has been the most popular acceptance criterion technique for ALNS (see e.g. Ropke and Pisinger (2006), Muller et al. (2012), Coelho et al. (2012)) and it is also used in this study. We accept the new solution  $s'$  over the current solution  $s$  if  $s'$  is better than  $s$ . Otherwise, it is accepted with the probability of  $e^{-(f(s')-f(s))/T}$ . The term  $f(s)$  represents objective function value (defined by (6.1)) for solution  $s$ . Here,  $T$  is the temperature which is updated by multiplying a cooling factor  $\mu$  ( $0 < \mu < 1$ ) at each iteration. Starting from initial temperature,  $T_{start}$ , the temperature is reduced. Instead of specifying a cooling rate parameter, we calculate  $T_{start}, T_{end}$  by using the initial solution obtained by the algorithm. We assume that the start temperature is  $\varphi\%$  of the initial solution while the ending temperature is  $\xi\%$  of initial solution. Then, the cooling factor becomes a parameter which guarantees the convergence from starting temperature to ending temperature after  $\varepsilon$  iterations. To further diversify the search, temperature  $T$  is reheated back to start temperature ( $T_{start}$ ) every  $\varepsilon$  iterations. This

means that depending on the maximum number of iterations ( $c_{max}$ ), the algorithm is reheated  $\lfloor \frac{c_{max}}{\epsilon} \rfloor$  times.

- Termination Criterion: The algorithm terminates once a number of iteration ( $c_{max}$ ) is reached. Selection of each parameter will be discussed in Section 3.6.1.

In the next subsection, we present the construction heuristic which generates the initial solution. Afterward, we introduce the operators used in the ALNS heuristic.

### 3.5.1 Construction Heuristic

Initially, we generate all assignments ( $\Omega$ ), then we select an assignment for each vessel. The initial solution is generated with a greedy approach. First, a random vessel order has been generated, and all assignments of each vessel are sorted in increasing cost order. The assignments of vessels in this order are placed into the solution one by one in a feasible way. If an assignment for a vessel cannot be inserted, the next assignment for that vessel is attempted. Further details of this method will be presented in Section 3.5.3.

### 3.5.2 Destroy Operators

In this study, destroy operators work at time, berth, QC and cost levels. Given the chosen destroy operator and a candidate solution  $X$ , the operator attempts to add the selected assignment to a destroy list, and it continues until  $\phi$  assignments are added to this list.

Once the destroy list, which is a list of assignments to be taken out of candidate solution, is compiled, the algorithm removes these assignments from the candidate solution. This results in a partial solution  $X_p$  in which  $\phi$  vessels are missing. This  $X_p$  will be the input for repair operator.

Some destroy operators select an assignment with a probability  $p$  ( $p$  is a parameter) in order to introduce some randomness in the selection. This is performed by drawing a random number  $q$  in  $(0, 1]$  and comparing it with  $p$ . If  $q \leq p$ , the current assignment is added to the destroy list; otherwise the next selected assignment is considered and the value of  $p$  is increased if necessary.

## Improved formulations and an adaptive large neighborhood search heuristic 100 for the integrated berth allocation and quay crane assignment problem

The number of assignments to be removed ( $\phi$ ) is determined by generating a random integer between  $\phi_u/2$  and  $\phi_u$ . The value  $\phi$  must satisfy  $2 \leq \phi \leq \phi_u$ , where  $\phi_u$  is the upper bound for the number of assignments that is intended to be destroyed. We determine  $\phi_u$  by dividing the size of the instance (e.g. 20, 30 or 40) to a parameter  $\phi_r$ .

If we choose to remove assignments that are not sharing same resources (like berth, QC), we might not gain anything when reinserting them. Therefore we now propose a set of removal operators based on different relatedness measures.

### 3.5.2.1 Shaw Removal

This removal operator was proposed by Shaw (1998). The operator returns a destroy list ( $D$ ) and a partial solution ( $X_p$ ). The general idea is to remove assignments that are related, as we expect it to be reasonably easy to shuffle related assignments around and thereby create new, perhaps better, solutions. The BACAP requires a special definition of relatedness. We define the relatedness measure  $M(i, j)$  for assignments of  $i$  and  $j$  to be:

$$M(i, j) = a|b_i - b_j| + b|s_i - s_j| + c|e_i - e_j|$$

$M(i, j)$  consists of three components: berthing position, berthing start time and berthing end time. The lower value of  $M(i, j)$  points a higher relatedness. The steps of the removal can be seen in Algorithm 3. The procedure initially chooses a random assignment to remove (line 1). Then, the relatedness vector is generated for all other assignments in the candidate solution with respect to selected assignment and these assignments are sorted in descending order of relatedness (line 2).

After we generate the sorted list, we use the determinism parameter  $p \in (0, 1]$  (line 7) to decide whether to put an assignment into the destroy list. In this operator, the parameter  $p$  is randomly generated between 0.6 and 1.0 which gives a relatively high probability of removal for the selected assignment, and  $p$  is updated to 1.0 if the number of remaining assignments in a candidate solution is equal to number of assignments that is needed to be put into destroy list (line 11). This increase is to ensure that exactly  $\phi$  assignments will be in the destroy list in the end. The weights of each factor ( $a, b, c$ ) are parameters for the algorithm which will be tuned in Section 3.6.1.1.

All of the assignments in the destroy list are removed from the candidate solution

### 3.5 Adaptive Large Neighborhood Search (ALNS) heuristic for BACAP 101

(line 13) and resource availabilities (berth, time, QCs) are updated.

---

**Algorithm 3:** Shaw Removal

---

```
Input      :  $X, p \in (0, 1], \phi \in \{1, \dots, N\}$ 
1 Select an assignment  $i$  in  $X$  randomly
2 Sort assignments with relatedness to  $i$  and generate sorted list
3 counter=0
4 while  $|D| < \phi$  do
5   | Select the assignment  $j$  in the sorted list, draw a random number
   |    $q$  in  $(0, 1]$ 
6   | counter++
7   | if  $q \leq p$  then
8   |   | Add  $j$  to the destroy list:  $D = D \cup \{j\}$ 
9   | else
10  |   | if  $(\phi - |D| = N - \text{counter})$  then
11  |   |   |  $p = 1$ 
12  | Go to next assignment in the sorted list
13 Remove all assignment in  $D$  from the candidate solution  $X$ 
14  $X_p \leftarrow X$ 
15 return  $X_p, D$ 
```

---

#### 3.5.2.2 Cost and Time-Relatedness Removal

This operator first sorts the assignments, which are in the candidate solution, in a decreasing cost order. Then it selects the highest cost assignment  $i$  and puts it in the destroy list. Afterward, it searches for time related assignments of the selected assignment  $i$ . Two assignments are time related if they occupy different berth positions at the same time. All time related assignments of  $i$  are added to the destroy list. If  $\phi$  assignments are not added to destroy list yet, then the next most costly assignment is added to destroy list and its time related assignments which are not in the destroy list are also added one by one. This procedure continues until  $\phi$  different assignments are collected in the destroy list. The motivation for using the time relatedness is to increase the likelihood of generating available QC capacity for high cost vessels.

### **3.5.2.3 Cost and Berth-Relatedness Removal**

This removal operator is very similar to the one presented in Section 3.5.2.2. The only difference is the use of berth-relatedness (assignments that share the same berthing positions at different times) instead of time-relatedness. This operator would bring flexibility to reinsert the vessels closer to their desired positions. Moreover, vessels that are placed closer to desired position require less QC capacity demand (i.e. shorter processing time).

### **3.5.2.4 Random Removal**

First an assignment is randomly selected and put into the destroy list. Then we randomly select new assignments. When a new assignment is randomly selected (after the first one), it is checked whether it is time-related with any of the assignments in the destroy list. If it is time-related, it is immediately put into the destroy list. Otherwise, like line 7 of Algorithm 3, the operator decides whether to put the assignment into the destroy list. In this operator, the parameter  $p$  is randomly generated between 0.6 and 1.0 once and updated as in line 10-11 of Algorithm 3.

## **3.5.3 Insertion Operators**

After the destroy operator is executed, we are left with a partial solution. In this partial solution, there are already  $N - \phi$  assignments with fixed berthing positions, start and end times, and QC plans. We differentiate between the insertion operator and the insertion list, because the insertion list is an input for the insertion operator. The insertion operator uses the order of vessels in the insertion list in order to insert them back into the partial solution. In this study, the insertion list is generated by randomly permuting the corresponding vessels of the assignments in the destroy list.

Since the problem is highly constrained we may end up in situations where there are some vessels that cannot be inserted into the current partial solution. This may happen in any insertion methods. We see such solutions as feasible but we add a high penalty for each unassigned vessel to the objective function in order to make it very attractive to reject such solutions. Two insertion operators have been suggested.

### 3.5 Adaptive Large Neighborhood Search (ALNS) heuristic for BACAP 103

#### 3.5.3.1 Basic Greedy Insertion

In order to apply this method, the insertion list ( $I$ ), all assignments of the vessels in insertion list ( $\Omega_k \quad \forall k \in I$ ), the partial solution ( $X_p$ ) and a determinism factor ( $p$ ) are required. All assignments of each vessel are sorted in increasing cost order. The operator tries to insert one assignment ( $i$ ) for each vessel ( $k$ ). Algorithm 4 illustrates the steps for the basic greedy insertion.

---

**Algorithm 4:** Basic greedy insertion

---

```

Input      :  $I, X_p, \phi = |I|, p \in (0, 1], \Omega_k : \forall k \in I$ 
1 for  $k = I_1 \rightarrow I_\phi$  do
2    $i = 0$ 
3   Draw a random number  $q$  in  $(0, 1]$ 
4   if  $(q > p)$  then
5      $i \rightarrow i + 1$ , go to 3
6   else
7     if  $(fOverlap(i, X_p) = false \wedge fQCapacity(i, X_p) = enough)$ 
8       then
9         Add  $i$  to partial solution  $X_p$ 
10        Make detailed QC_Assignment
11        Remove vessel  $k$  from insertion list:  $I = I - \{k\}$ 
12     else
13        $i \rightarrow i + 1$ , go to 3
14 return  $X$ 

```

---

The vessels in the insertion list are considered one by one. Algorithm 4 first selects the assignment of the first vessel in the insertion list. Here, we impose a randomness for accepting each assignment (line 4). There are three conditions to insert an assignment into the partial solution. The determinism parameter should support the insertion of the assignment, the assignment should not overlap with any of already inserted assignments in the partial solution, and there should be enough QC capacity to insert this assignment into the partial solution.

The control of overlapping ( $fOverlap(i, X_p)$ ) is fairly easy, because the partial solution includes all information about the berthing positions and the berthing intervals. The assignment  $i$  can be checked whether it overlaps with any assignments in the partial solution  $X_p$  in time and berth space. If it does not overlap, the function takes a "false" value.

The control of QC capacity availability for QC assignment ( $fQCapacity(i,$

$X_p$ )) requires some computational effort. Iris et al. (2015b) have proved that there is always an optimum  $f_1$  numbers of  $\hat{q}$  QCs and  $f_2$  numbers of  $\hat{q} + 1$  QCs which minimizes the cost of QC assignment plan and fulfill the QC requirement (i.e.  $\sum_{i \in V} (1 + \beta \Delta b_i) m_k$ ) with interference of  $\alpha$  (see Corollary 1 in Iris et al. (2015b)). The values of  $f_1$ ,  $f_2$  and  $\hat{q}$  are calculated by the algorithm in B.1. Note that,  $f_1, f_2$  presents for how many periods  $\hat{q}, \hat{q} + 1$  QCs will be used. Hence,  $f_1 + f_2$  should be equal to the processing time of that assignment. The values of  $\hat{q}, f_1, f_2$  are stored in the assignment. With the values of  $f_1, f_2$  and  $\hat{q}$ , we can control whether there is enough QC capacity for each period when the vessel will be at berth.

Since the number of QCs that will work on each vessel in each period is not kept in the assignment, we have to make a detailed QC plan with the available information (line 9). The detailed QC plan works with the principle of "as much as possible". We make this detailed QC plan period by period (from  $s_i \rightarrow s_i + process_i$ ). For each period of assignment's berthing interval, we first try to assign  $\hat{q} + 1$  QCs in that period. If there is not enough free QCs to assign  $\hat{q} + 1$  QCs for that period, we assign  $\hat{q}$  QCs instead. If we already assigned  $f_2$  numbers of  $\hat{q} + 1$  QCs for that vessel, we just assign  $\hat{q}$  QCs in the remaining periods. This procedure continues until the whole berthing interval is covered and detailed QC plan for that vessel is finalized. The aim is to assign  $\hat{q} + 1$  QCs as early as possible when there is free QC capacity. We also update free QC capacities every time a detailed QC plan is made, this is to evaluate next assignments accurately. Then we remove the corresponding vessel from insertion list. This procedure continues until all vessels in the insertion list have been inserted into the partial solution.

We illustrate the detailed QC plan with a small example. Assume that the vessel that will be inserted has  $\{r_k^{min}, r_k^{max}\} = \{3, 5\}$ . In this example, we disregard that the interference and a bad position can increase QC capacity demand which is 18 QC-hours. We assume that the attempted assignment does not overlap with any assignments in the partial solution. What is more, the assignment has a processing time of 5 periods. For the berthing interval of the assignment, we have the following number of QCs available:

Periods	1	2	3	4	5
QC Availability	5	3	5	5	4

The attempted assignment has a  $\hat{q}$  value of 3 ( $\lfloor \frac{18}{5} \rfloor$ ) which will be used for 2 periods, and  $\hat{q} + 1$  (4) QCs will be used for 3 periods ( $3 \cdot 2 + 4 \cdot 3 = 18$ ). The available QC capacities approve the use of QC plan which is stored in the assignment. The detailed QC plan starts with the first period, and assigns 4

### 3.5 Adaptive Large Neighborhood Search (ALNS) heuristic for BACAP 105

QCs to this period. The next period only has 3 available QCs, so 3 QCs are assigned to period 2, and so on. In the final period, since we have already assigned 4 QCs in 3 periods, we assign 3 QCs in that period. The complete algorithm results in following detailed QC plan:

Periods	1	2	3	4	5
QC plan	4	3	4	4	3

Two versions of this insertion operator are used in the ALNS algorithm. One version assumes a deterministic insertion ( $p = 1$ ), the other is the stochastic version where a randomly generated parameter  $p$  (between 0.25 and 1.0) is used.

#### 3.5.3.2 Smarter Greedy Insertion

The operator presented in the previous section can be improved by considering an alternative QC plan. The basic greedy operator uses the values of  $\hat{q}, \hat{q} + 1$  which form a minimum cost QC plan for a given assignment  $i$  of vessel  $k$  ( $\hat{q} = \left\lfloor \left( \frac{(1+\beta\Delta b_i)m_k}{process_i} \right)^{1/\alpha} \right\rfloor$ ). High values of  $\hat{q}, \hat{q}+1$  can violate the available QC number in a period, and this results in an infeasible QC plan. In such conditions, the basic greedy approach directly skips to the next assignment (which are ordered with increasing cost). The smarter greedy insertion reevaluates the assignment and makes an alternative QC plan for the considered assignment for each time period  $t$ .

We illustrate the conditions with another small example. Assume that the vessel that will be inserted has  $\{r_k^{min}, r_k^{max}\} = \{2, 6\}$ . In this example, we assume that the same conditions holds as the previous example ( $\alpha = 1.0, \beta = 0$ ), and QC capacity demand is 18 QC-hours. For the berthing interval of the assignment, we have the following number of QCs available:

Periods	1	2	3	4	5
QC Availability	7	6	2	3	4

In the basic greedy operator, the attempted assignment has a  $\hat{q} = 3$  ( $\lfloor \frac{18}{5} \rfloor$ ). We cannot make a detailed QC plan based on  $\hat{q}, \hat{q} + 1$ , since the period 3 has only 2 available QCs. The smarter greedy insertion detects this and tries to insert this assignment with an alternative QC plan. Let  $QC_{available}^t$  be the QC availability



## Improved formulations and an adaptive large neighborhood search heuristic 106 for the integrated berth allocation and quay crane assignment problem

in period  $t$ . The smarter greedy insertion first obtains the maximum possible QC number ( $QC_{possible}^t$ ) that can be assigned in time period  $t$  which is

$$QC_{possible}^t = \min\{QC_{available}^t, r_k^{max}\}$$

Then using  $QC_{possible}^t$ , the algorithm can determine the maximum number of effective QC hours that can be obtained in the berthing time interval, considering the interference factor. In our example, the calculation is  $6^1 + 6^1 + 2^1 + 3^1 + 4^1$  (21) which is adequate to meet the QC capacity demand (18). What is more, there are at least  $r_k^{min}$  (2) available QCs in each period during the berthing interval of the assignment. This means that an alternative QC plan is feasible for this assignment.

The alternative QC plan starts by sorting the periods in the berthing interval with increasing  $QC_{possible}^t$ . The QC plan is done in an iterative fashion, processing one time period at a time, in the sorted order. The algorithm maintains a variable  $QC_{needed}$  that indicates the number of QC hours that need to be provided in the remaining time periods. It starts out at the value of QC capacity demand (i.e.  $(1 + \beta\Delta b_i)m_k$ ) and the value is decreased after each assignment of QCs. By using  $QC_{needed}$  and the remaining number of time periods ( $p_{remain}$ ) until the berthing end time for that assignment, the algorithm compute a target value for the necessary number of QC in each remaining time period

$$Target = \left\lceil \left( \frac{QC_{needed}}{p_{remain}} \right)^{\frac{1}{\alpha}} \right\rceil$$

This value seeks to distribute the needed QC hours evenly over the remaining periods. It may happen that  $Target$  is less than  $r_k^{min}$  so we must use  $Target^* = \max(Target, r_k^{min})$ . Similarly it may not be possible to assign  $Target^*$  QCs in the period, because some QCs may be occupied by other vessels. Therefore the actual number assigned is obtained in the following way for each  $t$  being the time period under consideration:

$$QC_{assign}^t = \min\{Target^*, QC_{possible}^t\}$$

After the number of QCs to assign for each period is determined,  $QC_{needed}$  is updated.

$$QC_{needed} = QC_{needed} - (QC_{assign}^t)^\alpha$$

Due to the ordering of the time periods, the algorithm will find a feasible insertion if possible. Recalling the above example, we will show how the algorithm makes the updated QC plan. Sorting in the order of increasing  $QC_{possible}^t$ , it

yields the time period ordering 3,4,5,2,1 and the following table shows the assignment of the above mentioned variables in each iteration with  $QC_{assign}^t$  being the final assignment of QCs in period  $t$ .

Iteration	Time period ( $t$ )	$QC_{needed}$	$P_{remain}$	$Target^*$	$QC_{possible}^t$	$QC_{assign}^t$
1	3	18	5	4	2	2
2	4	16	4	4	3	3
3	5	13	3	5	4	4
4	2	9	2	5	6	5
5	1	4	1	4	6	4

The updated QC plan ( $QC_{assign}^t$ ) is potentially more costly compared to the QC plan stored in the assignment since it may use more QCs. Therefore it makes sense to investigate the following assignments to see if one is feasible and has lower cost. As soon as the algorithm reach an assignment whose stored cost is the same or higher than the assignment produced using the smarter greedy insertion described above the search can stop since the remaining assignments will have the same or a higher cost.

This procedure is applied for all vessels in the insertion list one by one, and continues until all vessels are added into the solution. Two versions of this insertion operator are used in the ALNS algorithm. One is a deterministic smarter insertion ( $p = 1$ ), the other is a stochastic version where a randomly generated parameter  $p$  (between 0.25 and 1.0) is used (See Appendix B.3 for the complete operator).

## 3.6 Computational Results

We compare our results to those that have been obtained in Meisel and Bierwirth (2009) and Iris et al. (2015b). All mathematical models and ALNS are run on a 32 core AMD Opteron at 2.8Ghz and 132Gb of RAM computer. All running times are measured in seconds.

### 3.6.1 Data and experimental settings

The benchmark, which includes 30 instances, has been obtained from Meisel and Bierwirth (2009). The size of an instance is defined with the number of vessels

## Improved formulations and an adaptive large neighborhood search heuristic 108 for the integrated berth allocation and quay crane assignment problem

(20, 30, or 40 vessels), 10 instances of each size are included in the benchmark. In each instance, there are three vessel types (namely Feeders, Medium, and Jumbo) and each of these vessel types differs in technical properties, and costs (See Meisel and Bierwirth (2009) for details). The quay wall length is 1000 meters which is 100 berthing positions (10-meter segmented) and 10 QCs are available on the quay. The planning horizon is to 168 working hours (one week), and all planning and assignment operations are based on working hours. The interference coefficient ( $\alpha$ ) is 0.9, and the increase in the QC-hours needed due to berthing deviation ( $\beta$ ) is 0.01.

All models are solved by using CPLEX 12.6.1 solver. We have run the mathematical models under the conditions presented in Meisel and Bierwirth (2009) and Iris et al. (2015b). All models are run with a computational time limit of 10 hours. All cores are used when solving the models. In order to have a fair comparison, the models presented by Meisel and Bierwirth (2009) and Iris et al. (2015b) are rerun. Each model has been initiated with a warmstart solution which is the result of the ALNS heuristic for one run.

The ALNS algorithm is implemented in C++, and it uses only one thread. The heuristic is attempted 10 times on each instance and we report best, average and worst solution for each instance. The termination criterion is a maximum number of 270000 iterations for each run.

### 3.6.1.1 Parameter tuning

First, a set of representative tuning instances is generated. The tuning set consists of 15 instances and it is generated randomly. The tuning instances are different from the instances used in the following sections. We first introduce the parameters that will be tuned. We start with the removal parameters. The Shaw removal has three parameters ( $a, b, c$ ). The number of vessels to destroy is managed by  $\phi_r$ , this parameter  $\phi_r$  will be also tuned. The SA is controlled by three parameters,  $\varphi, \xi, \epsilon$ . The weight-adjustment is made with four parameters;  $\sigma_1, \sigma_2, \sigma_3$  and  $\eta$ . Finally, the update interval of the weights ( $\delta$ ) should also be tuned.

We use a hierarchical procedure for tuning the parameters. First, a phase of parameter settings, which consist of the two reasonable values for each parameter, is decided. We analyze all combinations of these parameters in this phase. Since there are 12 parameters to tune, the first phase runs  $2^{12}$  different parameter settings. This phase results in an initial parameter setting for the ALNS. The first parameter setting results are improved in the second phase. In second phase, each parameter is tested for four values, while

the rest of the parameters are fixed in the best value obtained in the first phase. For the parameters, which we believe are more important  $(\phi_r, \varphi, \xi, \varepsilon)$ , we impose five values to test in the second phase. For each parameter setting, we run the ALNS ten times. This process continues until all parameters have been tuned. The setting that shows the best average objective value is chosen. The complete tuning phase results in a parameter vector of:  $(a, b, c, \phi_r, \varphi, \xi, \varepsilon, \sigma_1, \sigma_2, \sigma_3, \eta, \delta) = (0.01, 2.0, 0.01, 4, 0.05, 0.0005, 30000, 2, 5, 10, 0.8, 1000)$ .

Because of the fact that the parameter tuning aims at obtaining the best parameter setting with respect to objective function, we may give a compromise on the runtime of the algorithm. We note that some of the parameters directly affect the runtime of the algorithm. For these parameters, we have made an ad-hoc analysis of different values where we analyze the performance of the algorithm with respect to both runtime and objective function. The results show that  $\phi_r$  which controls how many vessels to remove/insert from/to solution at each iteration is the most influential parameter on the runtime. Different values of  $\phi_r$  are tested on the tuning instances, average objective function and average runtime for each setting are presented in Table 3.2. The results show that  $\phi_r$  can be updated to 6 instead of 4, because the average objective with 4 is slightly better compared to 6 case. But, there is a meaningful decrease in the computational time.

**Table 3.2:** Parameter  $\phi_r$  vs. Solution Quality

$\phi_r$	3	4	5	6
Avg. (1)	176.41	175.88	176.35	175.89
Avg. $T_k$	203.5	150.2	113.0	103.0

### 3.6.2 Results for improved formulations

Before presenting the results for BACAP, BACAP+ models, let us analyze the performance of valid inequalities and variable fixing methods. Table 3.3 presents the average performance of the linear programming (LP) relaxation LBs for each instance size. There are two main parts in Table 3.3. The first part presents the results which include CPLEX presolver and cutting planes. These options are then disabled, and the LBs are reevaluated. For each column in Table 3.3, we present the gap between the LB and the upper bound obtained by the warmstart solution. We also report the CPU times in the parenthesis, these are the times to obtain the LBs. If additional time is required to generate a parameter that will be used in an inequality, this time is also added to CPU time. The first column

## Improved formulations and an adaptive large neighborhood search heuristic 110 for the integrated berth allocation and quay crane assignment problem

---

( $N$ ) specifies the instance size, the second column ( $G_R$ ) presents the average gap of LBs for the original formulation (BACAP without enhancements).

The columns  $V_1, V_2, \dots, \Theta$  present the results of the formulation with the active inequalities.  $V_1$  refers to the valid inequalities (3.22), and  $V_2$  refers to (3.23).  $V_3$  includes the valid inequalities about start and end decomposition (3.24-3.37).  $V_4$  covers the set partitioning inequalities (3.39-3.42).  $V_5$  have the variable fixing inequalities (3.43-3.45). The families of inequalities which are about objective function are clustered into a larger group. As a result,  $V_6$  has the objective function dependent inequalities (3.47-3.48). The symbol  $\Theta$  corresponds to all sets of inequalities. Note that valid inequalities (3.38) have been considered as part of the original formulation.

The first important observation from Table 3.3 is that the performances of the inequalities are not that strong when presolver and cutting planes are disabled. But when these two CPLEX options are activated, CPLEX generates very useful cuts and the average integrality gap is reduced for each set of instances. Another important observation is that including all inequalities results in better LBs and this proves the contribution of the inequalities.

Let us now analyze the details of each set of inequalities with respect to the gaps. The performances are evaluated for the case in which CPLEX presolver and cutting planes are kept active. The results show that for small scale instances, the performance of the inequalities is erratic. For medium and large scale instances, the contribution is more clear. In general, by adding all inequalities, the gap is reduced by 27% in average (case  $\Theta$ , last column). If we look at one inequality class at a time, we can observe that valid inequalities  $V_3$  and  $V_1$  perform better than the rest in the average. Individual results for each instance show that each family of inequalities has significant contribution on different instances. Although the gaps seem to be similar for  $V_1, V_2, V_5, V_6$ , each family of inequalities contributes to different instances.

It should be noted that inequalities  $V_4, V_5$  require to run the two preprocessing methods (Iris et al. (2015b)) to obtain the bounds used in these constraints. For this reason, the computational time to generate these bounds (e.g.  $MinS_i, MaxE_i, MaxB_i$ , etc.) are added to LP relaxation time. In Iris et al. (2015b), the authors also propose two probing methods to further improve the LBs. However, the high computational times to obtain these bounds with the probing methods did not justify their use in the BACAP+ formulation.



The LB obtained with one type of inequality is sometimes worse than the one which is obtained with the original BACAP formulation (e.g. Instance 3,  $V_1$ ). This is explained by the fact that CPLEX uses a heuristic separation procedure for the cuts, which leads to the generation of a different set of inequalities. With the help of Table 3.3, we justify to use all inequalities in the BACAP+ model.

The results of BACAP, BACAP+ and GSPP models are presented in Table 3.4. In this table, the column "N" shows the number of vessels (instance size), "#" indicates the instance ID. The columns named "Z" show the best upper bounds obtained, while "BLB" report the best LBs found within the time limit. The optimality gap ( $G = \frac{Z - BLB}{Z}$ ) is calculated between the upper and lower bounds. The columns " $T_c$ " show the time spent to solve the mathematical model, while " $R_{LB}$ " reports the LBs which are used in Table 3.3.

In Table 3.4, "BACAP Model" results are obtained from Meisel and Bierwirth (2009) paper. The "BACAP with warmstart" presents the rerun results for Meisel and Bierwirth (2009) model with a warmstart solution. The "BACAP+" indicates results for default CPLEX settings, while "BACAP+ (Cuts&Opt. Emph)" presents the BACAP+ results with CPLEX option of *emphasize optimality* and *aggressive cut generation*. Finally, the "GSPP" presents the results of reruns of the GSPP models which are presented in Iris et al. (2015b). All mathematical models analyzed in this paper (BACAP, BACAP+, GSPP) start with the same warmstart for each instance.

The original results in Meisel and Bierwirth (2009) presented optimal solutions for four instances, and the model cannot generate any integer solutions for almost all medium and large scale instances.

The results show that for small and medium scale instances BACAP+ formulations can solve all instances with less than 2% of optimality gap. For large scale instances ( $N = 40$  vessels), no instance can be solved to optimality within the 10 hour time limit. However, each instance results in a better optimality gap compared to the BACAP with warmstart. For instances which both the BACAP+ model and the original BACAP model find the optimal solutions, BACAP+ always needs less computational effort. Additionally, almost all upper and lower bounds in the literature have been improved by BACAP+ models with the warmstart. The BACAP+ model also remains competitive with the GSPP model. In average, the BACAP+ model performs better in small, medium and large scale instances. However, the number of instances which are solved to optimality is higher for the GSPP model. In Table 3.4, we write gaps or computational times, which are meaningfully better compared to other results, in bold font.

The comparison of different CPLEX settings is also relevant. The results show

**Table 3.4:** Comparison results of Model by Meisel and Bierwirth (2009), GSPP model by Iris et al. (2015b), BACAP+

		BACAP Model				BACAP with warmstart				BACAP+ with warmstart				BACAP+ (Cuts&Opt, Enpl.)				CSPP with warmstart							
N	#	Z	BLB	Cl%	Z	BLB	Cl%	T <sub>C</sub>	R <sub>LB</sub>	Z	BLB	Cl%	T <sub>C</sub>	R <sub>LB</sub>	Z	BLB	Cl%	T <sub>C</sub>	R <sub>LB</sub>	Z	BLB	Cl%	T <sub>C</sub>	R <sub>LB</sub>	
20	1	84.1	84.0	0.12	84.1	84.1	0.0	453	70.2	84.1	84.1	0.0	14	77.8	84.1	84.1	0.0	28	77.2	84.1	84.1	0.0	101	83.3	
	2	53.9	53.9	0.0	53.9	53.9	0.0	3	50.3	53.9	53.9	0.0	2	51.0	53.9	53.9	0.0	5	53.6	53.9	53.9	0.0	1	53.94	
	3	77.4	75.2	2.93	76.3	76.0	0.32	36000	66.2	76.3	76.1	0.25	36000	68.3	76.3	76.1	0.18	36000	69.0	76.3	76.3	0.0	245	75.3	
	4	76.2	75.8	0.52	76.2	75.9	0.37	36000	66.0	76.2	76.2	0.0	510	68.2	76.2	76.2	0.0	8072	70.3	76.2	76.2	0.0	390	74.4	
	5	56.8	56.8	0.00	56.8	56.8	0.0	12	49.7	56.8	56.8	0.0	10	52.6	56.8	56.8	0.0	13	51.2	56.8	56.8	0.0	32	56.81	
	6	57.6	57.6	0.00	57.6	57.6	0.0	2	57.6	57.6	57.6	57.6	0.0	2	56.0	57.6	57.6	0.0	4	56.2	57.6	57.6	0.0	1	57.64
	7	68.0	67.5	0.74	68.0	68.0	0.0	445	67.5	68.0	68.0	0.0	58	61.8	68.0	68.0	0.0	129	65.1	68.0	68.0	0.0	12	67.1	
	8	56.1	56.1	0.00	56.1	56.1	0.0	26	51.0	56.1	56.1	56.1	0.0	13	51.8	56.1	56.1	0.0	22	51.9	56.1	56.1	0.0	21	55.9
	9	75.1	75.0	0.13	75.1	75.1	0.0	324	62.0	75.1	75.1	75.1	0.0	68	67.0	75.1	75.1	0.0	101	66.2	75.1	75.1	0.0	158	73.2
	10	90.9	88.2	3.06	89.3	88.9	0.36	36000	65.6	89.3	89.1	0.18	36000	72.8	89.3	89.1	0.18	36000	73.6	89.3	87.8	1.68	36000	83.0	
30	11	X	137.7	-	140.4	134.8	3.93	36000	95.8	138.8	139.6	0.09	36000	106.5	138.8	139.6	0.08	36000	108.1	138.8	138.4	0.22	36000	133.8	
	12	81.8	81.4	0.49	81.8	81.8	0.0	478	74.3	81.8	81.8	0.0	35	74.3	81.8	81.8	0.0	74	74.3	81.8	81.8	0.0	41	81.81	
	13	104.9	100.9	3.96	102.4	101.8	0.55	36000	88.4	102.4	102.1	0.28	36000	88.6	102.4	102.1	0.29	36000	89.3	102.4	102.4	0.0	792	100.8	
	14	X	96.8	-	99.1	98.3	0.79	36000	81.3	99.1	98.7	0.33	36000	85.2	99.1	98.8	0.24	36000	85.7	99.1	99.1	0.0	2558	97.5	
	15	X	136.9	-	150.4	135.4	9.94	36000	91.9	150.4	149.5	0.55	36000	104.6	150.4	148.4	1.27	36000	106.8	150.4	148.5	1.20	36000	136.8	
	16	X	106.2	-	113.8	110.7	2.64	36000	86.1	113.8	113.2	0.47	36000	90.5	113.8	113.4	0.31	36000	91.3	113.9	111.8	1.78	36000	107.8	
	17	X	99.6	-	102.6	102.0	0.58	36000	74.8	102.6	102.2	0.39	36000	87.1	102.6	102.3	0.24	36000	88.7	102.6	102.6	0.0	403	100.7	
	18	X	117.8	-	121.9	118.9	2.44	36000	93.6	121.9	121.5	0.32	36000	104.8	121.9	121.4	0.42	36000	104.0	121.9	120.5	1.13	36000	117.4	
	19	X	156.4	-	165.2	143.3	13.2	36000	108.9	165.2	162.1	1.91	36000	129.8	165.2	163.3	1.11	36000	127.0	165.2	162.2	1.82	36000	158.8	
	20	X	125.6	-	132.5	121.6	8.21	36000	97.5	131.5	131.0	0.36	36000	103.6	131.5	131.1	0.29	36000	104.8	131.6	129.7	1.41	36000	128.6	
40							4.22					0.47					0.42					0.75			
	21	X	165.7	-	201.9	148.4	26.4	36000	100.0	201.9	178.7	11.48	36000	137.6	201.9	181.3	10.15	36000	137.9	202.0	179.4	11.60	36000	178.0	
	22	X	159.6	-	170.9	152.2	10.91	36000	110.9	170.9	163.8	4.12	36000	127.1	170.9	165.3	3.25	36000	122.9	171.0	166.8	2.44	36000	163.1	
	23	X	185.0	-	239.5	162.2	32.21	36000	100.3	234.2	199.8	14.68	36000	122.2	234.2	205.0	12.47	36000	121.1	264.41	196.71	25.60	36000	196.4	
	24	X	224.1	-	284.0	193.9	31.37	36000	119.3	282.5	239.0	15.39	36000	152.1	282.3	243.5	14.04	36000	152.0	302.21	245.41	18.80	36000	239.4	
	25	X	133.3	-	148.7	126.8	14.71	36000	104.1	148.9	141.0	5.29	36000	110.7	148.7	143.7	4.79	36000	111.4	148.1	143.4	3.13	36000	139.9	
	26	X	201.3	-	288.6	166.0	30.40	36000	96.4	230.7	208.5	9.61	36000	135.0	229.3	211.6	7.71	36000	135.1	245.01	208.81	14.80	36000	208.0	
	27	X	172.2	-	183.8	149.2	18.79	36000	97.8	181.9	173.9	5.41	36000	135.9	181.9	175.8	4.40	36000	139.8	188.0	174.7	7.41	36000	173.9	
	28	X	211.7	-	260.9	161.8	37.89	36000	102.7	262.1	204.1	22.11	36000	156.2	262.1	218.9	16.47	36000	156.1	315.21	215.31	31.70	36000	215.3	
	29	X	180.3	-	207.0	152.6	26.24	36000	123.5	207.0	193.5	6.48	36000	126.7	207.1	194.9	5.87	36000	126.7	208.8	185.3	11.24	36000	185.5	
30	X	170.1	-	190.9	163.6	24.05	36000	117.3	185.8	174.2	6.21	36000	131.0	185.1	179.5	3.01	36000	131.7	187.81	175.01	6.80	36000	173.9		
							26.05					10.07					8.21					13.35			

† represents that CPLEX solver has faced a memory overflow while the instance was running using 5 threads. Hence, † are results of Iris et al. 2015b), ‡ represents that the instance is solved to optimality in the root node



that the average gap is reduced with "emphasizing optimality" and "aggressive cut generation" settings. With these settings, CPLEX obtains the same or better optimality gaps compared to the default settings in 27 out of 30 instances. BACAP+ with the new settings also obtains a better root node LB compared to the default settings in many instances. However, the number of optimal solutions does not change with the new settings.

### 3.6.3 Results for ALNS

In Table 3.5, we present the computational performance of ALNS compared to the best known upper bounds and two state-of-art heuristics: Squeaky Wheel Optimization (SWO) and Tabu Search (TS) from Meisel and Bierwirth (2009). The ALNS algorithm is run 10 times for each instance. The average, best, and worst objective values, all relevant gaps, average run times, and the time for generating all assignments are reported for each instance.

In Table 3.5, column " $Z^*$ " shows the best upper bound known in the literature. Then next seven columns report the results for the ALNS. The table also includes the results of SWO and TS of Meisel and Bierwirth (2009).

The average gap is calculated using the average cost and  $Z^*$  (average gap =  $\frac{\text{average cost} - Z^*}{\text{average cost}}$ ), while the best gap is calculated in a similar way (best gap =  $\frac{\text{best cost} - Z^*}{\text{best cost}}$ ). The "ALNS-[23] gap" is the gap between the best result among SWO and TS and the average ALNS result ( $\frac{\text{average cost} - \text{best SWO/TS cost}}{\text{average cost}}$ ). The negative value of this gap shows how better the ALNS performs compared to Meisel and Bierwirth (2009) best results. While,  $T_{k1}$  is the average computational time of ALNS and  $T_{k2}$  is the computational time for SWO or TS heuristics.  $T_c$  is the time to generate the set of assignments ( $\Omega$ ) which is considered as part of the ALNS.

Table 3.5: ALNS Results

#	Z*	ALNS										SWO			TS	
		Average		Best	Worst	Average	Best	ALNS-[23]	T <sub>k1</sub>	T <sub>c</sub>	Cost	T <sub>k2</sub>	G (%)	Cost	T <sub>k2</sub>	G (%)
		Cost	Cost	Cost	G (%)	G (%)	G (%)	G (%)								
1	84.1†	84.8	84.1	85.1	0.83	0.00	-0.35	13.8	0.6	85.1	11	1.18	85.1	14	1.18	
2	53.9†	53.9	53.9	53.9	0.00	0.00	0.00	2.5	0.5	53.9	4	0.00	53.9	8	0.00	
3	76.3†	78.2	77.4	78.4	2.38	1.42	0.97	30.1	0.4	77.4	11	1.42	77.4	17	1.42	
4	76.2†	79.7	77.9	87.7	4.39	2.18	2.26	18.6	0.6	79.7	8	4.39	77.9	12	2.18	
5	56.8†	57.0	56.8	57.7	0.32	0.00	0.32	8.9	0.4	56.8	10	0.00	56.8	15	0.00	
6	57.6†	57.6	57.6	57.6	0.00	0.00	0.00	2.0	0.5	57.6	4	0.00	57.6	7	0.00	
7	68†	69.0	68.3	69.2	1.38	0.44	0.07	6.1	0.5	68.9	17	1.31	68.9	24	1.31	
8	56.1†	58.0	56.1	59.9	3.28	0.00	3.28	6.8	0.5	57.0	8	1.57	56.1	13	0.00	
9	75.1†	75.3	75.3	75.4	0.29	0.27	-0.24	7.6	0.5	75.9	18	1.05	75.5	14	0.53	
10	89.3	92.2	90.1	94.6	3.19	0.89	-0.82	13.9	0.6	94.6	10	5.60	93	10	3.98	
			G <sub>av</sub>		1.60	0.52	0.55	10.9			10.1	1.65		13.4	1.06	
11	139.8	146.6	143.3	147.8	4.61	2.44	-0.85	68.6	0.8	147.8	51	5.41	149.5	61	6.48	
12	81.8†	82.4	82.4	82.4	0.73	0.73	-0.12	8.1	0.7	83.3	17	1.80	82.5	36	0.85	
13	102.4†	105.4	104.7	105.6	2.81	2.20	0.82	14.1	0.7	105.7	53	3.12	104.5	41	2.00	
14	99.1†	104.1	103.3	105.6	4.76	4.07	-1.68	47.5	0.7	105.8	22	6.33	113.2	41	12.45	
15	150.4	154.2	152.4	157.0	2.49	1.31	-2.07	45.8	0.8	159	57	5.41	157.4	79	4.44	
16	113.8	116.5	116.4	117.6	2.32	2.23	-1.72	46.2	0.9	118.5	40	3.97	119.5	51	4.76	
17	102.6†	104.0	103.5	104.2	1.31	0.87	-0.23	24.5	0.7	104.5	38	1.82	104.2	41	1.53	
18	121.9	125.1	124.6	125.6	2.59	2.17	-0.32	34.1	0.8	125.5	20	2.87	131.2	46	7.07	
19	165.2	171.3	170.1	173.1	3.54	2.88	-1.48	58.8	1.0	173.8	27	4.95	173.8	41	4.95	
20	131.5	136.2	135.3	136.3	3.44	2.81	0.73	48.7	0.8	135.2	58	2.74	138.3	93	4.91	
			G <sub>av</sub>		2.86	2.17	-0.70	39.9			38.4	3.85		53	4.94	
21	201.9	208.6	204.6	210.3	3.19	1.32	-3.09	133.6	1.2	215	311	6.09	226.7	209	10.93	
22	170.9	179.5	178.5	180.3	4.81	4.26	0.41	83.4	0.9	178.8	163	4.42	183.4	165	6.81	
23	234.2	251.4	244.6	263.0	6.84	4.25	-5.13	163.7	1.0	273.9	315	14.49	264.3	373	11.39	
24	282.5	290.2	285.9	294.9	2.73	1.26	-12.54	143.7	1.1	326.6	325	13.56	342.2	351	17.44	
25	148.1	154.8	152.9	155.9	4.32	3.14	-0.01	87.1	0.8	155.1	206	4.51	154.8	140	4.33	
26	229.3	249.1	245.9	256.7	7.93	6.75	-4.23	183.8	1.0	260.4	130	11.94	259.6	298	11.67	
27	183.8	191.9	189.1	193.5	4.21	2.80	-4.65	145.3	0.9	200.8	209	8.47	215.8	282	14.82	
28	260.9	272.9	263.4	286.3	4.39	0.95	-1.88	208.1	1.2	286.2	373	8.84	294.3	109	11.34	
29	207	215.5	209.9	222.9	3.92	1.38	-1.83	71.3	1.0	219.4	202	5.65	223.4	175	7.34	
30	185.1	202.0	194.9	207.8	8.35	5.03	-19.28	128.0	1.1	240.9	209	23.16	254.7	395	27.32	
			G <sub>av</sub>		5.07	3.11	-5.52	134.9			244.3	10.11		249.7	12.38	

$G_{av} = [\frac{\sum_{i \in n} G_i}{n}]$ : Average gap or computational time for each instance size, † represents that the  $Z^*$  is proved to be the optimal solution.

## Improved formulations and an adaptive large neighborhood search heuristic 116 for the integrated berth allocation and quay crane assignment problem

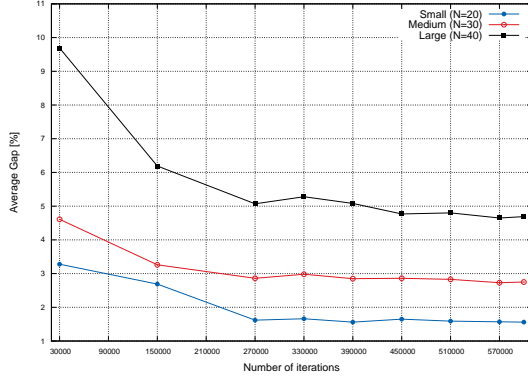
The results show that the average gaps of the ALNS from the best-known solution for small, medium, and large scale instances are 1.6%, 2.86%, and 5.07% respectively. These results support that the ALNS heuristic obtains good results in short computational times. If we look at individual instance, there are five instances for which ALNS obtains the optimal solutions. For the majority of the instances, the ALNS is able to find the best known heuristic solution. The improvement from average solution to best solution is 1.27% in average.

The results also show that average gaps of ALNS are mostly better than the heuristics presented in Meisel and Bierwirth (2009). In particular, the ALNS results are better than the best results of SWO/TS heuristics for medium and large scale instances. For small scale instances, the best results of SWO/TS is better than the ALNS. The average ALNS results are better or equal compared to best of SWO/TS results in 22 of 30 instances. When the best solution of ALNS is compared to the best solutions of SWO/TS, ALNS is better for 29 of 30 instances. The results also show that ALNS is stronger as the instance size increases.

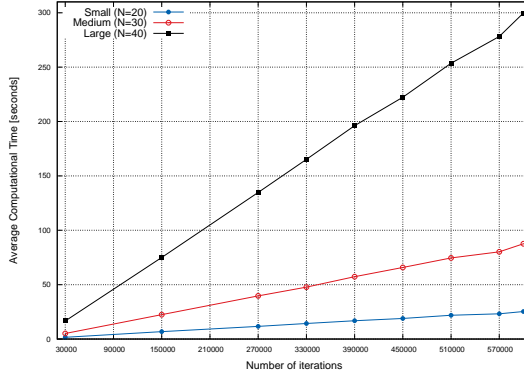
Showing that ALNS mostly obtains better upper bounds compared to SWO and TS heuristics, we should also emphasize that these bounds are mostly obtained in comparable computational times (Note that SWO and TS results were obtained using a 2.4 GHz Pentium IV which must be assumed to be slower compared to our computer). We also report the time to generate the set of assignments which is at most 1.2 seconds. In average, ALNS consistently outperforms SWO both with respect to optimality gap and computational time. The same comment holds for TS except TS obtains a better gap for small scale instances.

The computational time and the optimality performance tradeoff of ALNS is strongly affected by the number of iterations and the parameter  $\phi_r$  (which controls how many vessels to be removed/reinserted at every iteration). For this reason, we do further tests that aim at pointing out the performance of the heuristic for different values of the two parameters. For the remaining parameters, the parameter tuning results are used. All further performance analyses are done using the average results over 10 runs.

Figure 3.3 illustrates the average gap (a) and the average computational time (b) for different number of iterations. The results are clustered for each instance size. The average gap of all instance sizes converge strongly after approximately 450000 iterations. The average computational time linearly increases as the number of iteration increases. The best average gaps obtained are approximately 4.7%, 2.7%, and 1.5% for large, medium and small scale instances. Figure 3.3 supports the selection of 270000 iterations as the termination criterion, since the gap improvements are minimal after this value and the computational times are also reasonable.



(a)

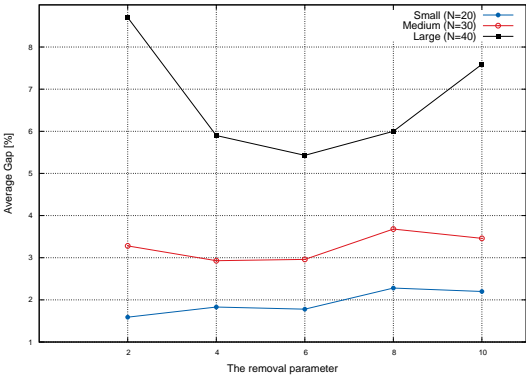


(b)

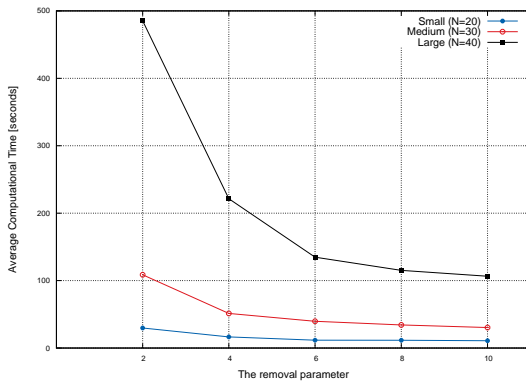
**Figure 3.3:** Analysis of number of iteration

Figure 3.4 illustrates the average gap (a) and the average computational time (b) for different values of the  $\phi_r$  parameter. This removal parameter adversely affects the number of vessels that will be removed/inserted at every iteration. The higher value of  $\phi_r$  results in less number of vessels to be removed, and vice versa. Figures 3.4a and 3.4b are drawn for each instance size. Figure 3.4a shows that the least average gap is obtained for  $\phi_r$  values of 6 for large, 4 for medium and 2 for small scale instances. Figure 3.4b points out that the average computational time decreases as the value of  $\phi_r$  increases. With the help of the two subfigures, we can state that the tradeoff between computational time and optimality performance of ALNS has a break-even point at  $\phi_r$  of 6.

Finally, we also identify how well each insertion operator performs if they are used as the only insertion operator. We report the average increase in the



(a)



(b)

Figure 3.4: Analysis of  $\phi_r$

average solution cost (%) for different combinations of the insertion operators in Table 3.6.

Table 3.6: Impact of insertion operators on the ALNS performance

Greedy insertion		Smarter Greedy insertion		Increase in the average cost (%)
deterministic	stochastic	deterministic	stochastic	
1	0	0	0	2.81
0	0	1	0	1.70
1	1	0	0	2.96
0	0	1	1	1.85

The first observation from Table 3.6 is that each combination increases the average objective value. This means that the use of all insertion methods together has a contribution to the ALNS algorithm. The results also show that the use of deterministic smarter greedy insertion as the only insertion method, yields the least amount of increase in the average objective value and the smarter greedy insertion consistently outperforms the basic greedy insertion method in both deterministic and deterministic+stochastic versions.

### 3.7 Conclusions

In this chapter, we have proposed a number of families of valid inequalities and variable fixing methods in order to improve the state-of-the-art BACAP formulations. Additionally, we have presented an ALNS heuristic to solve the problem more efficiently. The results show that both inequalities and the heuristic improve the state-of-the-art results. The inequalities and ALNS can easily be used for related research problems with some modifications.

The minimum and maximum processing time and bounds presented by Iris et al. (2015b) have been used efficiently to formulate the inequalities. We should note that the average gaps of BACAP+ model are consistently better than gaps of GSPP and BACAP models for each instance size. Compared to other formulations in the literature, the BACAP+ model improves (or at least find the same) upper bounds for all, except two instances. The best LBs obtained by BACAP+ model are mostly better than GSPP models of Iris et al. (2015b).

The ALNS heuristic uses different destroy and repair methods which diverse the search space efficiently. Comparative tests on instances have shown that our heuristic can produce high quality solutions compared to other heuristics presented in the state-of-the-art both with respect to computational time and solution quality.

The problem formulation can also be extended to cover more complex QC planning. In this respect, a future research direction could be the modification of the ALNS to solve the BACASP.

**Improved formulations and an adaptive large neighborhood search heuristic  
120 for the integrated berth allocation and quay crane assignment problem**

---

## CHAPTER 4

# A two-stage stochastic programming approach to berth and quay crane scheduling problem under uncertainty

---

This work focuses on three important problems that deal with bottleneck operations on quayside, namely Berth Allocation Problem, Quay Crane Assignment and Scheduling Problems. The state-of-art studies mostly rely on deterministic vessel arrivals, known vessel loads and they mostly assume that uniform quay crane operations are maintained. However, in reality, these parameters are mostly random. In this study, we model the integrated berth and quay crane scheduling problem under uncertainty as a two-stage stochastic programming problem, we also present a decomposition algorithm to solve it. Results show that our formulation and decomposition approach efficiently solve the problem and the value of stochastic information is important for the problem. <sup>1</sup>

---

<sup>1</sup>C. Iris, J.G. Jin, D.H. Lee, S. Ropke, "A two-stage stochastic programming approach to berth and quay crane scheduling problem under uncertainty", 2016, under review



## **4.1 Introduction**

Accounting for about 23% of 6.8 billion tons of dry cargo trade, World containerized trade is transported via container terminals. The recent statistics also show that the global containerized trade have increased by 5.3% and reached 171 million TEUs in 2014 (UNCTAD (2014)). These statistics reveal the increasing volumes in container terminals and such an increase breeds congestions and uncertainties in the operations. In this respect, the efficient use of resources in the terminal has attracted many researchers in the recent years (see following review for application of operations research in terminal operations: Stahlbock and Voß (2007)).

The performance of a container terminal heavily relies on how efficient its quayside resources, which are mainly berth and quay cranes (QCs), are used. In the literature, each of the resources has been considered in a number of problem definitions. The usage of the berthing area is among of the important problems. The optimization of vessels' berthing positions and berthing time is formulated by Berth Allocation Problem (BAP), while QC management is mainly formulated by Quay Crane Assignment and Scheduling Problem (QCASP). In a version of QCASP, only the number of QCs that will be assigned to each vessel along its berthing interval is decided. In another version, it is also investigated which QCs will be working on each vessel for each period and the complete schedule of QCs is also made by considering many operational constraints. In conventional studies, berth and QC related problems are addressed separately. However, the number of QCs assigned to each vessel directly affects the berthing interval which is a major component of BAP. As a result, integration of these problems became very relevant.

The quayside operations are interface between the liner shipping company and terminal operators. The flow of information between these two parties might fail or some delays might occur. There are many sources of uncertainties in the planning of quayside operations which is already a complicated problem composing of many subproblems such as BAP, QCASP, etc. The state-of-art studies mostly rely on forecasted, deterministic vessel arrivals, and they mostly assume that processing times are known for each vessel. In the daily operations, the decisions about the berthing order of vessels and QC assignment are actually made prior to knowing the exact vessel arrival times which are affected by some unpredictable or uncontrollable factors (weather, congestion, lack of information flow, exact container load on the vessel, breakdowns of QCs, etc.).

In this chapter, we focus on modeling the integrated berth and quay crane scheduling problem under uncertainty. It is assumed that the number of QCs assigned to a vessel cannot change over time. Additionally, it is assumed that

the berthing positions are known for each vessel in advance. This means we do not determine the berthing position for each vessel, but we schedule the berthing order with respect to given berth allocation. The problem is formulated as a two-stage stochastic programming problem and resulting formulation is solved both by using a Benders decomposition variant approach and a black-box integer programming (IP) solver.

Observations from terminal operations show that the exact arrival time of each vessel and the processing time (if  $q$  number of QCs are assigned to given vessel) are two stochastic parameters. We assume that these stochastic parameters realize simultaneously and they are unknown by the time of planning. Although, there are scenarios with certain realization probabilities, and these scenarios hold information about the stochastic parameters.

The contribution of this chapter is mainly three-fold. We formulate, to the best of our knowledge, the first traditional two-stage stochastic programming model that focuses on berth and QC scheduling problem. The stochasticity is reflected by parameters which strongly affect the decisions made by practitioners. Secondly, we propose a stage-wise decomposition approach to solve this problem. We present valid inequalities to strengthen the formulation. The computational results reveal that the decomposition approach performs better as the number of scenarios increases and it is competitive with the deterministic equivalent of the stochastic programming model. We also provide results that show the value of information for this problem, and it helps to obtain important savings for terminal operators.

In section 2, we present a literature review on the relevant problems. In section 3, we introduce the problem in the mathematical form and present the two-stage stochastic programming model. The model will be explained stage-wise. In section 4, we present the L-shaped method, which is based on Benders decomposition, to solve the problem. We also suggest some valid inequalities for master problem and accelerating strategies in this section. Computational results are discussed in section 5. Finally, we present the conclusion and the future research potential in the last section.

## 4.2 Related literature

The integrated problems in the quayside operations have been attracting many studies recently. The papers in this field have been extensively reviewed in Bierwirth and Meisel (2010), Bierwirth and Meisel (2014), Carlo et al. (2013), Iris et al. (2015b).

The integrated Berth Allocation and quay Crane Assignment and Scheduling Problem (BACASP) has been first studied by Park and Kim (2003). The authors solve the integrated berth allocation and QC assignment problem in the first stage by using a subgradient optimization strategy. The QC assignment in the first stage only deals with the number of QCs to assign each vessel in each period. The results from the first stage are then used in the second stage to solve the specific QC assignment and scheduling problem. A dynamic programming technique is used to solve this problem. Imai et al. (2008) present one of the first mathematical models which formulates a complete BACASP. The authors propose a genetic algorithm to solve the problem. Meisel and Bierwirth (2013) also focus on BACASP in which three problems are deeply integrated. The authors present various heuristic methods and test them for different integration levels of the subproblems. Turkogullari et al. (2014) formulate a novel model for the discrete BACASP. The authors emphasize that the model is not effective for large scale instances. Hence, they propose a post-processing cutting plane algorithm over the results of a BACAP solution. Experiments show that the largest instances can be solved to optimality with this method. Their main assumption is that the number of QCs assigned to a vessel cannot be changed over time (see also Iris et al. (2015b)). Li et al. (2015a) have focused on a BACASP in which QC coverage ranges are also considered. The authors present a mathematical model which has many similar considerations with Meisel and Bierwirth (2009) model. They also propose a heuristic algorithm which is based on spatio-temporal conflicts analysis.

Liu et al. (2006) is one of the first papers on Berth and quay Crane Scheduling Problem (BCSP). In their berth-level model, authors assume that the berthing position for each vessel is known in advance and they determine the berthing start/end times, the number of QCs to assign to each vessel and which specific QC will be assigned. Chen et al. (2012) use the berth-level model of Liu et al. (2006) and propose a combinatorial Benders cut algorithm to solve the problem. In this chapter, we build our two-stage stochastic programming model by modifying the berth-level model of Liu et al. (2006). There are two main reasons for this. Firstly, the problem has many realistic merits, secondly the decomposition of the problem yields a linear programming (LP) model in the subproblem which makes the complete problem easier to solve by a stage-wise decomposition approach.

As many literature reviews note, the number of papers that focus on the uncertainty in quayside problems is limited. One of the first studies that focuses on the uncertainties in quayside operations is Moorthy and Teo (2006). The authors have noted the need for a model that incorporates uncertainty due to vessel arrivals. The research papers in the field can be divided into two mainstream categories. There are papers that consider some stochastic parameters and solve the problem in a static (proactive) way (e.g. le Han, qiang Lu, and feng

Xi (2010), Karafa, Golias, Ivey, Saharidis, and Leonardos (2013), Golias et al. (2014), Rodriguez-Molins, Ingolotti, Barber, Salido, Sierra, and Puente (2014a), etc.). Most of the proactive studies focus on generating robust plans by considering time buffers (le Han et al. (2010), Golias et al. (2014), Rodriguez-Molins et al. (2014a)). While others present dynamic (reactive) disruption recovery methods (e.g. Li, Jin, and Lu (2015b), Umang, Bierlaire, and Erera (2016)).

The number of studies that approach to the problem stage-wise is also very limited. Zhen, Lee, and Chew (2011b) formulate a two-stage stochastic programming model to solve berth allocation problem under uncertain arrival time or operation time of vessels. The authors assume that a baseline schedule is generated in the first stage. They try to minimize the cost of this baseline schedule and expected recovery costs of all scenarios with given probabilities. The problem is solved with a simulated annealing based algorithm. In their study, the authors also present a dynamic recovery technique. Unlike the traditional two-stage stochastic programming problems, they allow the first stage decision variables (baseline plan) to change after the realization of the scenarios. This is one of the main differences of our work from Zhen et al. (2011b). In a later study, Zhen (2015) focuses on tactical berth allocation problem under uncertainty. The study proposes both stochastic programming and robust optimization formulations to cope with randomness in the arrival and processing times. Heuristic methods are presented to solve the problems.

To the best of our knowledge, this study is the pioneering study that focuses on a traditional two-stage stochastic programming formulation for a container terminal problem under uncertainty. What is more, it is the first study that uses a stage-wise decomposition approach to solve such a problem. The different examples of stage-wise decomposition methods can be found in Pishvaei, Razmi, and Torabi (2014), Oliveira, Nunes, Blajberg, and Hamacher (2016), Agra, Christiansen, Delgado, and Hvattum (2015), Kiya and Davoudpour (2012), Cardona-Valdes, Alvarez, and Ozdemir (2011).

### 4.3 Problem definition and mathematical formulation

The problem studied in this paper aims at finding the berthing start and end times for each vessel and how many QCs will be assigned to each vessel. Apart from the number of QCs, the problem aims at determining which specific QCs will be assigned to each vessel. It is assumed that the berthing position of each vessel is known and the berthing order of vessels that share the same berthing position at some point is given. A continuous berthing partitioning is used for

the quay. This means a vessel or a QC can be positioned at any point along the quay. The berthing start and end times are also continuous variables which allows the vessel to berth at any time along the planning horizon.

Every berthed vessel has an upper and lower limit on the number of assignable QCs. These values are determined by the size of vessel and contract between ship operator and terminal. A limited number of QCs are available along the berth and it bounds the maximum number of QCs that can be assigned to all vessels at anytime.

The stochastic problem is formulated in the two stages. In the first stage, a berthing order and QC assignments for all vessels are decided without any information on the vessel arrival and processing times. The second stage (recourse) decisions involve determining the exact berthing start and end times of each vessel and scheduling of QCs for all vessels by considering the non-overtaking constraints. After determining first stage variables, the terminal can prepare its quay configuration with the decided settings (e.g. pre-marshalling, moving QCs to initial positions, etc.). This will prevent additional setup costs of QCs and the cost of changing or updating the yard configuration and picking up orders. This property is the main motivation to formulate the problem as a two-stage stochastic problem.

The problem uses discrete probability distributions of scenarios (with certain probabilities). Each scenario holds stochastic parameters which are obtained through a simple sampling approach. We let the probability of each scenario  $\omega$  be  $\rho^\omega$ . We try to minimize the cost of weighted completion time of all vessels and the cost of QC operations. Since the actual arrival and processing times realize in the second stage, there is no cost component in the first stage. The solution to the complete problem includes fixed first stage decisions and decisions for each second stage scenario problem.

The indices of QCs increases one by one along the berth, this increase has same direction with the increase in the position value along the berth. Since the berthing positions of vessels are known in advance, the precedence relationships can be defined for all vessels. In the Table 6.1, we first present the sets and parameters which are related to both stages, and then we list the first stage variables followed by second stage variables in Table 4.2.

Table 4.1: Notation

Parameters and sets:	
$V$	the set of vessels that are going to be berthed in the planning horizon, $V \in \{1, 2, \dots, N\}$ where $N$ is the number of vessels to berth
$\Phi$	the set of quay cranes available along the berth, $\Phi \in \{1, 2, \dots, Q\}$ where $Q$ is the number of available (overall) QCs
$\mathbb{T}$	the set of vessel pairs $(i, j)$ such that vessel $j$ is the next vessel that berths along some berth position occupied by vessel $i$
$\mathbb{T}^*$	the set of vessel pairs $(i, j)$ that include all the pairs in $\mathbb{T}$ and pairs $(i, j)$ such that $\{(i, i_1), (i_1, i_2), \dots, (i_e, j)\} \in \mathbb{T}$ for some vessels $i_1, i_2, \dots, i_e$
$\mathbb{U}$	the set of all ship pairs $(i, j)$ such that both $(i, j) \notin \mathbb{T}^*$ and $(j, i) \notin \mathbb{T}^*$
$a_i^\omega$	the arrival time of vessel $i$ under scenario $\omega$ , (Stochastic parameter)
$p_{iq}^\omega$	the processing time under scenario $\omega$ when $q$ quay cranes are assigned to vessel $i$ , (Stochastic parameter)
$d_i$	the expected departure time of vessel $i$
$l_i$	the (mid-point) position of vessel $i$
$\tau$	the total preperation time on the quayside for berthing vessel to depart and replacing new vessel into quay
$cost_i^1$	the cost of one time-unit processing time for vessel $i$
$cost^2$	the cost of QC operations for one time-unit
$r_i^{min}$	the minimum number of QCs agreed to serve vessel $i \in V$ simultaneously
$r_i^{max}$	the maximum number of QCs allowed to serve vessel $i \in V$ simultaneously
$R_i$	the set of number of QCs assignable to vessel $i \in V$ , where $R_i \in \{r_i^{min}, \dots, r_i^{max}\}$
$M_1, M_2$	two large positive numbers

We now introduce the decision variables of each stage.

**Table 4.2:** Decision variables

First-stage variables:	
$r_{iq} \in \mathbb{B}$	1 if $q$ QCs are assigned to work on vessel $i$ , 0 otherwise
$z_{ki} \in \mathbb{B}$	1 if QC $k$ is assigned to vessel $i$ , 0 otherwise
$y_{ij}^{ss} \in \mathbb{B}$	1 if berthing start time of vessel $i$ is before berthing start time of vessel $j$ where $(i, j) \in \mathbb{U}$ , 0 otherwise
$y_{ij}^{sc} \in \mathbb{B}$	1 if berthing start time of vessel $i$ is before berthing end time of vessel $j$ where $(i, j) \in \mathbb{U}$ , 0 otherwise
$y_{ij}^{cs} \in \mathbb{B}$	1 if berthing end time of vessel $i$ is before berthing start time of vessel $j$ where $(i, j) \in \mathbb{U}$ , 0 otherwise
$y_{ij}^{cc} \in \mathbb{B}$	1 if berthing end time of vessel $i$ is before berthing end time of vessel $j$ where $(i, j) \in \mathbb{U}$ , 0 otherwise
Second-stage variables:	
$s_i \in \mathbb{R}^+$	berthing start time of vessel $i \in V$
$c_i \in \mathbb{R}^+$	berthing end time (time when the handling ends) of vessel $i$
$xs_{ki} \in \mathbb{R}^+$	the position of quay crane $k$ at the berthing start time of vessel $i$
$xc_{ki} \in \mathbb{R}^+$	the position of quay crane $k$ at the berthing ending time of operations of vessel $i$

### 4.3.1 Master problem and recourse function

Figure 4.1 shows an example master problem solution in a time/quay diagram. In this example, seven vessels are berthed. Each vessel is represented by a rectangle showing the time and space occupied by the vessel with vessel index. The smaller rectangles in gray indicate the vessels' QC assignments with each grey one representing one QC. The number in each of grey rectangles represents which QC is assigned to that vessel. Each vessel has an upper and lower limit on the number of assignable QCs ( $r_i^{min}, r_i^{max}$ ). A limited number of QCs are available in the berth and this determines the maximum number of QCs that can be assigned at The number and index of QCs does not change throughout the vessels stay at the berth. We first clarify sets of vessels with this example and then denote different variables.

- $\mathbb{T} = \{(1, 4), (1, 5), (2, 6), (3, 5), (4, 7), (6, 7)\}$
- $\mathbb{T}^* = \{(1, 4), (1, 5), (1, 7), (2, 6), (2, 7), (3, 5), (4, 7), (6, 7)\}$
- $\mathbb{U} = \{(1, 1), (1, 2), (1, 3), (1, 6), (2, 1), (2, 2), (2, 3), (2, 4), (2, 5), (3, 1), (3, 2), (3, 3), (3, 4), (3, 6), (3, 7), (4, 2), (4, 3), (4, 4), (4, 5), (4, 6), (5, 2), (5, 4), (5, 5), (5, 6), (5, 7), (6, 1), (6, 3), (6, 4), (6, 5), (6, 6), (7, 3), (7, 5), (7, 7)\}$

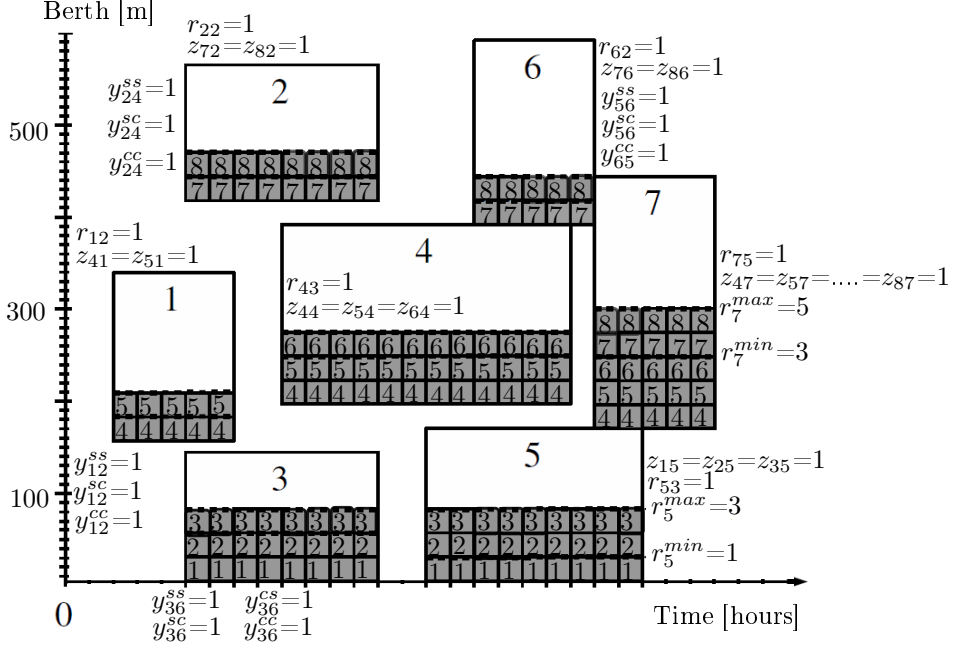


Figure 4.1: Example of a master problem solution

The variables in the master problem are on vessel sequencing and QC assignment. In Figure 4.1, vessel 4 has 3 QCs assigned ( $r_{43} = 1$ ) where these QCs are 4, 5 and 6 (i.e.  $z_{44} = z_{54} = z_{64} = 1$ ). In this figure, we denote these two variables for all vessels except vessel 3. Regarding vessel sequencing variables, vessel 1 has an earlier berthing start time compared to vessel 2 ( $y_{12}^{ss} = 1$ ) and its start time is also earlier than vessel 2's end time ( $y_{12}^{sc} = 1$ ). Finally, the end time of vessel 1 is earlier than vessel 2's start time ( $y_{12}^{cc} = 1$ ). Giving as an example, vessel 1's end time is not earlier than vessel 2's start time ( $y_{21}^{cs} = 0$ ), hence  $y_{21}^{cs}$  takes a value of 1. Figure 4.1 denotes various examples of these sequencing variables. Figure 4.1 does not hold any specific start or end time for each vessel, because these times can be anything that hold the sequencing conditions. We remind the reader that berth and time axes are continuously partitioned.

We now present the following two-stage stochastic programming model:

$$\min \quad 0 + \mathfrak{J}(\chi) \quad (4.1)$$



subject to

$$\sum_{q \in R_i} qr_{iq} = \sum_{k \in \Phi} z_{ki} \quad \forall i \in V \quad (4.2)$$

$$y_{ij}^{ss} + y_{ji}^{ss} = 1 \quad \forall (i, j) \in \mathbb{U} \quad (4.3)$$

$$y_{ij}^{sc} + y_{ji}^{cs} = 1 \quad \forall (i, j) \in \mathbb{U} \quad (4.4)$$

$$y_{ij}^{cc} + y_{ji}^{cc} = 1 \quad \forall (i, j) \in \mathbb{U} \quad (4.5)$$

$$\sum_{q \in R_i} r_{iq} = 1 \quad \forall i \in V \quad (4.6)$$

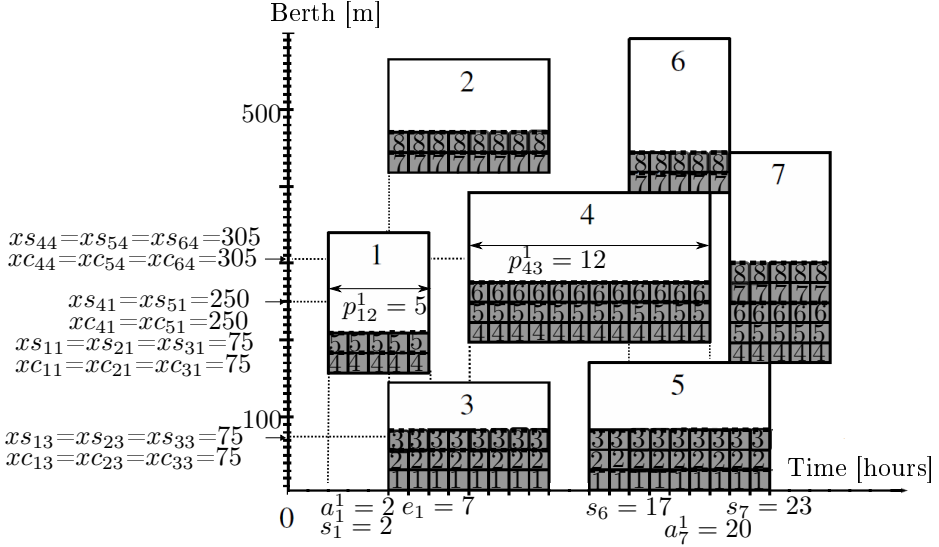
$$r_{iq}, z_{ik}, y_{ij}^{ss,sc,cs,cc} \in \{0, 1\} \quad \forall (i, j) \in \mathbb{U}, \forall i \in V, \forall q \in R_i, \forall k \in \Phi \quad (4.7)$$

where

$$\mathfrak{J}(\chi) = E_{\omega}[\mathfrak{J}(r; y; z; \omega)] \quad (4.8)$$

The objective of the two-stage stochastic model includes the expected cost of second stage problem. There are no costs imposed from the first-stage variables (master problem), this is because the arrival and processing time, which realize in the second stage, determine the overall cost. Constraint (4.2) links the number of QCs and the assignment of specific QCs for each vessel. Constraints (4.3)-(4.5) are vessel sequencing constraints. They take care of vessel scheduling in the quay. Since vessels  $i$  and  $j$  share same berthing position at different time units, either vessel  $i$  or vessel  $j$  starts or ends berthing earlier (4.3 and 4.5). This property also guarantees that either berthing start of vessel  $i$  is earlier than berthing end time of vessel  $j$ , or vice-versa (4.4). Constraint (4.6) ensures that at most one QC number can be assigned to each vessel. This is because we assume that the number of QCs that work on a vessel does not change over time. And finally, constraint (4.7) sets the domain of decision variables. Here the function  $\mathfrak{J}(\chi)$  accounts for the expected value of second-stage problem  $\mathfrak{J}(r; y; z; \omega)$  for the fixed values of  $r, y, z$  variables for scenario  $\omega$ . By removing  $\mathfrak{J}(\chi)$  from the above formulation, we are left with the master problem for the two-stage stochastic programming problem.

Once the first-stage decisions are made, the second stage recourse actions must be taken with respect to different realizations of stochastic parameters. Each scenario,  $\omega \in S$ , is associated with a processing time ( $p_{iq}^{\omega}$ ), an arrival time ( $a_i^{\omega}$ ) and a probability  $\rho^{\omega}$ . Figure 4.2 shows an example recourse problem solution in a time/quay diagram for scenario 1. In this example, the solution of master problem (Figure 4.1) is inherited with QC assignment and vessel sequencing. Figure 4.2 illustrates variables regarding the berthing start/end times and QC positions.



**Figure 4.2:** Example of recourse problem solution for scenario 1

In Figure 4.2, QCs 1, 2 and 3 are processing at vessel 3 initially. For this reason  $xs_{i3}$  and  $xc_{i3}$  are equal to 75 for  $i = 1, 2, 3$ . This value is the mid-point position of berthing vessel 3. Then QCs 4 and 5 are operating on vessel 1 ( $xs_{41} = xc_{41} = xs_{51} = xc_{51} = 250$ ). Since vessel 1 starts berthing earlier than vessel 3, and ends operations later than vessel 3's berthing has started, vessel 1 and 3 are at the berth at the same time. For this reason, QCs 1, 2 and 3 hold the positions of mid-point of vessel 3 for vessel 1 (i.e.  $xs_{11} = xc_{11} = xs_{21} = xc_{21} = xs_{31} = xc_{31} = 75$ ). This is in order to prevent the overtaking of QCs. With respect to berthing start and end times, vessel 1 has been illustrated. It is shown in Figure 4.2 that berthing start time is 2 ( $s_1 = 2$ ), while the arrival time for the same vessel is also 2 under scenario 1 ( $a_1^1 = 2$ ). For vessel 1 under scenario 1, when 2 QCs are assigned 5 hours are needed to load/unload required containers ( $p_{12}^1 = 5$ ), while the berthing end time for vessel 1 ( $e_1$ ) is 7. As the x-axis of Figure 4.2 shows, the variables of start/end time can take any positive real number. Figure 4.2 holds some illustrative examples of the remaining variables.

For each scenario  $\omega$ , the second stage recourse problem (RP) is in essence of solving a berth and QC scheduling problem. Given the berth and QC assignment from the first-stage problem, in the second-stage, the berthing start/end time depending on vessel's arrival/processing time and specific QC assignment/schedule

is decided. The recourse problem is:

$$[RP] :: \mathfrak{J}(\chi) = \min \sum_{i \in V} cost_i^1 \frac{c_i}{d_i} + cost^2 \sum_{i \in V} \sum_{q \in R_i} q r_{iq}^{\hat{\omega}} p_{iq}^{\omega} \quad (4.9)$$

subject to

$$s_i \geq a_i^{\omega} \quad \forall i \in V \quad (4.10)$$

$$c_i \geq s_i + \sum_{q \in R_i} p_{iq}^{\omega} r_{iq}^{\hat{\omega}} \quad \forall i \in V \quad (4.11)$$

$$s_j \geq s_i + M_1(y_{ij}^{\hat{s}s} - 1) \quad \forall (i, j) \in \mathbb{U} \quad (4.12)$$

$$c_j \geq s_i + M_1(y_{ij}^{\hat{s}c} - 1) \quad \forall (i, j) \in \mathbb{U} \quad (4.13)$$

$$s_j \geq c_i + M_1(y_{ij}^{\hat{c}s} - 1) \quad \forall (i, j) \in \mathbb{U} \quad (4.14)$$

$$c_j \geq c_i + M_1(y_{ij}^{\hat{c}c} - 1) \quad \forall (i, j) \in \mathbb{U} \quad (4.15)$$

$$xs_{ki} \geq l_i + M_2(\hat{z}_{ik} - 1) \quad \forall i \in V, k \in \Phi \quad (4.16)$$

$$xs_{ki} \leq l_i + M_2(1 - \hat{z}_{ik}) \quad \forall i \in V, k \in \Phi \quad (4.17)$$

$$xc_{ki} \geq l_i + M_2(\hat{z}_{ik} - 1) \quad \forall i \in V, k \in \Phi \quad (4.18)$$

$$xc_{ki} \leq l_i + M_2(1 - \hat{z}_{ik}) \quad \forall i \in V, k \in \Phi \quad (4.19)$$

$$xs_{kj} \leq l_i + M_2(3 - \hat{z}_{ik} - y_{ij}^{\hat{s}s} - y_{ji}^{\hat{s}c}) \quad \forall (i, j) \in \mathbb{U}, k \in \Phi \quad (4.20)$$

$$xs_{kj} \geq l_i + M_2(\hat{z}_{ik} + y_{ij}^{\hat{s}s} + y_{ji}^{\hat{s}c} - 3) \quad \forall (i, j) \in \mathbb{U}, k \in \Phi \quad (4.21)$$

$$xc_{kj} \leq l_i + M_2(3 - \hat{z}_{ik} - y_{ij}^{\hat{s}c} - y_{ji}^{\hat{c}c}) \quad \forall (i, j) \in \mathbb{U}, k \in \Phi \quad (4.22)$$

$$xc_{kj} \geq l_i + M_2(\hat{z}_{ik} + y_{ij}^{\hat{s}c} + y_{ji}^{\hat{c}c} - 3) \quad \forall (i, j) \in \mathbb{U}, k \in \Phi \quad (4.23)$$

$$xs_{ki} - xs_{k-1,i} \geq 0 \quad \forall i \in V, k \in \Phi \setminus \{1\} \quad (4.24)$$

$$xc_{ki} - xc_{k-1,i} \geq 0 \quad \forall i \in V, k \in \Phi \setminus \{1\} \quad (4.25)$$

$$s_j - c_i \geq \tau \quad \forall (i, j) \in \mathbb{T} \quad (4.26)$$

$$\hat{z}_{ik} = z_{ik}^* \quad \forall i \in V, k \in \Phi \quad (4.27)$$

$$y_{ij}^{\hat{s}s} = y_{ij}^{ss*} \quad \forall (i, j) \in \mathbb{U} \quad (4.28)$$

$$y_{ij}^{\hat{s}c} = y_{ij}^{sc*} \quad \forall (i, j) \in \mathbb{U} \quad (4.29)$$

$$y_{ij}^{\hat{c}s} = y_{ij}^{cs*} \quad \forall (i, j) \in \mathbb{U} \quad (4.30)$$

$$y_{ij}^{\hat{c}c} = y_{ij}^{cc*} \quad \forall (i, j) \in \mathbb{U} \quad (4.31)$$

$$r_{iq}^{\hat{\omega}} = r_{iq}^* \quad \forall i \in V, q \in R_i \quad (4.32)$$

$$y_{ij}^{\hat{s}s}, y_{ij}^{\hat{s}c}, y_{ij}^{\hat{c}s}, y_{ij}^{\hat{c}c} \in \mathbb{R}^+ \quad \forall i, j \in V, i \neq j \quad (4.33)$$

$$xc_{ki}, xs_{ki}, s_i, c_i \in \mathbb{R}^+ \quad \forall i \in V, k \in \Phi \quad (4.34)$$

The objective of [RP] is to minimize the total cost of the weighted completion time of all vessels and QC operations (4.9). Constraint (4.10) ensures that the berthing start time for vessel  $i$  should be greater than the actual arrival time of vessel  $i$ . Constraint (4.11) sets berthing end time when the processing time of vessel  $i$  is set where  $q$  QCs are assigned to that vessel. The following constraints (4.12)-(4.15) ensure that the berthing start and end times are set properly with respect to the vessel berthing order. Constraints (4.16)-(4.23) determine QC positions along the berth. If a QC is assigned to a vessel, it moves to the berthing position of that vessel (4.16-4.17) and works there until the berthing end time of that vessel (4.18-4.19). These constraints hence guarantee that the QC operations are non-preemptive. Constraints (4.20)-(4.23) link the berthing orders and specific QC assignments with the positions of QCs along the berth. It must be also guaranteed that QCs cannot over-take each other when they move. Constraints (4.24)-(4.25) ensure that the position of QC  $k$  is always bigger than position of QC  $k - 1$  at any time by assuming that positions are incremental along the berth. For a ship pair  $(i, j)$  that uses the same berthing position consecutively, there should be at least  $\tau$  time difference between the berthing end time of vessel  $i$  and berthing start time of vessel  $j$  (4.26). Constraints (4.28)-(4.31) fix the first stage decision variables in the second stage. Finally, the domains of variables are set. We should note that  $z_{ik}^*, r_{iq}^*, y_{ij}^{ss*, sc*, cs*, cc*}$  are parameters which are inherited from master problem. The values of  $Big - M$  should be discussed. For each given subproblem  $\omega$ ,  $M_1 = \max_{i \in V, q \in R_i} \{a_i^\omega + p_{iq}^\omega\}$  and  $M_2 = \max_{i \in V} \{l_i\}$ .

### 4.3.2 Valid inequalities for master problem

In this section, we present families of valid inequalities to tighten the master problem which is an IP formulation. We use these inequalities both in our stage-wise decomposition approach and deterministic equivalent of the stochastic programming model.

We firstly improve the master problem with respect to the number of available QCs in each period along the berth. If two vessels are at different berth positions at the same time-unit, they have to share the pool of QCs which is a limited number. Therefore we can formulate a valid inequality stating that such vessels can at most use  $Q$  QCs together. Let us first define a new binary decision variable  $O_{ij}$  which is 1 if vessel  $i$  and  $j$  are at the quayside at the same time where  $(i, j) \in \mathbb{U}$ , 0 otherwise. Constraints (4.35)-(4.36) ensure that the variable  $O_{ij}$  is set accurately. If the berthing start time of vessel  $i$  is earlier than vessel  $j$  and the berthing start time of vessel  $j$  is earlier than the berthing end time of vessel  $i$ , these two vessels should be together at the berth at some time (4.35). It must be ensured that if vessel  $i$  is at berth together with vessel  $j$ , vessel  $j$

should also be together with vessel  $i$  at some time (4.36). That should also be ensured that all  $(i, j) \in \mathbb{T}^*$  should take  $O_{ij} = 0$ , because these vessels are known to be sharing the same berthing positions which means they cannot be at the berth at the same time (4.37).

$$O_{ij} \geq y_{ij}^{ss} + y_{ji}^{sc} - 1 \quad \forall (i, j) \in \mathbb{U} \quad (4.35)$$

$$O_{ij} = O_{ji} \quad \forall (i, j) \in \mathbb{U} \quad (4.36)$$

$$O_{ij} = 0 \quad \forall (i, j) \in \mathbb{T}^* \quad (4.37)$$

In order to formulate the valid inequality, we apriori generate all combinations of vessels for different number of elements ranging from 2 to  $N$ . Let us call the set of vessel combinations  $C_p$  where  $p \in \{1, 2, \dots, ((\binom{N}{2}) + (\binom{N}{3}) + \dots + (\binom{N}{N}))\}$ . Each of these vessel combinations holds vessels as elements. Giving an example, let us assume that we plan 4 vessels. Then, we can write the vessel combination set as;  $C_1 = \{1, 2\}$ ,  $C_2 = \{1, 3\}$ ,  $C_3 = \{1, 4\}$ ,  $C_4 = \{2, 3\}$ ,  $C_5 = \{2, 4\}$ ,  $C_6 = \{3, 4\}$ ,  $C_7 = \{1, 2, 3\}$ ,  $C_8 = \{1, 2, 4\}$ ,  $C_9 = \{1, 3, 4\}$ ,  $C_{10} = \{2, 3, 4\}$ ,  $C_{11} = \{1, 2, 3, 4\}$ . The set points out which vessel combinations can share QC capacity. Constraint (4.38) ensures that all vessels that overlap in time share at most  $Q$  available QCs.

$$\sum_{i \in C_p} \sum_{q \in R_i} qr_{iq} \leq Q \left\{ |C_p|(|C_p| - 1) - \sum_{i \in C_p} \sum_{j \in C_p \setminus \{i\}} O_{ij} + 1 \right\} \quad \forall p \in \{1, 2, \dots, |C_p|\} \quad (4.38)$$

The left-hand side (LHS) of constraint (4.38) is the total number of QCs required if  $q$  QCs are assigned for each vessel through its stay at berth. The right-hand side (RHS) fixes the upper bound (UB) to  $Q$  only if all of the vessels in the given combination are at the quayside at the same time. If the value of  $\sum_{i \in C_p} \sum_{j \in C_p \setminus \{i\}} O_{ij}$  is equal to  $|C_p|(|C_p| - 1)$ , this means that all vessels in  $C_p$  are at the quayside together at some time point. Therefore the total QC number that is assigned to them in total cannot be larger than  $Q$ . Constraint (4.38) is formulated for each of the combinations in  $C_p$ .

There could be situations in which only one vessel is at berth. Hence we cannot use the variable  $O_{ij}$ . We can formulate a valid inequality which ensures that at most  $Q$  QCs can be assigned to each vessel (4.39).

$$\sum_{q \in R_i} qr_{iq} \leq Q \quad \forall i \in V \quad (4.39)$$

The next set of valid inequalities deals with the assignment of specific QCs to each vessel. If we know that two vessels are at the quay at the same time, depending on the berthing position of each vessel, the specific QCs that might be assigned to each vessel are limited. This property is a result of the assumption that the number of QCs that work on vessel and QCs themselves cannot be changed after the vessel is berthed. Constraint (4.40) guarantees that if  $O_{ij}$  is one and the berthing position of vessel  $i$  is smaller than vessel  $j$ , the indices of all QCs assigned to vessel  $j$  should be higher than vessel  $i$ . (See Chen et al. (2012) for a version of this constraint).

$$z_{ki} - \sum_{k_0 \in \Phi, k < k_0} z_{k_0j} + O_{ij} \leq 1 \quad \forall (i, j) \in \mathbb{U}, k \in \Phi, \quad l_i < l_j \quad (4.40)$$

Vessels that are the berth at the same time cannot use the same QC. This property is formulated as the valid inequality (4.41) where if  $O_{ij}$  is one then  $z_{ki}$  and  $z_{kj}$  cannot be one together.

$$z_{ki} + z_{kj} + O_{ij} \leq 2 \quad \forall i, j \in V, k \in \Phi \quad i \neq j \quad (4.41)$$

We can also formulate inequalities that use the information stored in the scenario tree. The minimum and maximum arrival time of vessel  $i$  in the scenario can be extracted. Let us call these parameters  $EAT_i, LAT_i$  (earliest arrival time and latest arrival time) for each vessel  $i$  which is  $EAT_i = \min_{\omega \in S} \{a_i^\omega\}, LAT_i = \max_{\omega \in S} \{a_i^\omega\}$ . We can also extract the minimum and maximum processing time in the scenario tree. Let us call these parameters  $SPT_i, LPT_i$  (shortest processing time and longest processing time) for each vessel  $i$  which is  $SPT_i = \min_{\omega \in S, q \in R_i} \{p_{iq}^\omega\}, LPT_i = \max_{\omega \in S, q \in R_i} \{p_{iq}^\omega\}$ . We introduce a new positive continuous decision variable  $ss_i$  which presents the earliest possible berthing start time for each vessel  $i$  in the master problem. We can formulate the following inequalities which tighten relationship between ordering of the vessels in set  $\mathbb{U}$ .

$$ss_i \geq EAT_i \quad \forall i \in V \quad (4.42)$$

$$(LAT_i + LPT_i)y_{ij}^{cs} \leq ss_j \quad \forall (i, j) \in \mathbb{U} \quad (4.43)$$

Chen et al. (2012) formulate some valid inequalities for the problem introduced by Liu et al. (2006). We now briefly explain ones that our valid inequalities

cannot dominate. If QCs  $k$  and  $k^1$  (say,  $k < k^1$ ) are assigned to vessel  $i$  together, the QCs located at the interval between these two quay cranes (i.e., quay crane  $k_0$ ,  $k < k_0 < k^1$ ) should also be assigned to vessel  $i$  due to the physical constraint of non-crossing. In the other words, the index of QCs assigned to a vessel should be consecutive. The constraint (4.44) presents the related link.

$$z_{k_0,i} \geq z_{k,i} + z_{k^1,i} - 1 \quad \forall i \in V, \quad k, k_0, k^1 \in \Phi \quad k < k_0 < k^1 \quad (4.44)$$

Finally, Chen et al. (2012) discuss that if the berthing start time of vessel  $i$  is after the berthing end time of vessel  $j$ , the berthing start time of vessel  $j$  is also earlier than berthing start time of vessel  $j$ , such comments also hold for the berthing end time. The constraints that address these discussions can be found in (4.45).

$$y_{ij}^{sc} \geq y_{ij}^{ss}, y_{ij}^{cc} \geq y_{ij}^{cs} \quad \forall i, j \in V, \quad i \neq j \quad (4.45)$$

## 4.4 Solution method: Integer L-shaped method

The two-stage formulation can be transformed into a one compact mixed-integer programming model which is often referred as the Deterministic Equivalent Formulation (DEF). In this version of formulation, each second stage variable inherits a new indice for each scenario and all second-stage constraints are rewritten for each scenario. The master problem variables and constraints remain the same. Then the objective function is the expected cost of second stage variables. (i.e.  $\min \sum_{\omega \in S} \rho^\omega \{ \sum_{i \in V} cost_i^1 \frac{c_i(\omega)}{d_i} + cost^2 \sum_{i \in V} \sum_{q \in R_i} q r_{iq} p_{iq}^\omega \}$ ). The complete DEF will not be presented here, but results of it will be communicated in section 4.5. When the number of scenarios increases, solving the DEF becomes burdensome very rapidly. Hence we also propose an integer L-shaped based decomposition approach to solve the problem.

The integer L-shaped algorithm is an extension of traditional Benders' decomposition (L-shaped) that solves the two-stage stochastic problems which have integer/binary variables in the master problem in an iterative fashion (Birge and Louveaux (2011), Laporte and Louveaux (1993)). The main idea of the L-Shaped algorithm is to approximate the non-linear term in the objective (i.e., the recourse function). Since the second-stage problem is an LP,  $\mathfrak{J}(\chi)$  is convex in  $\chi$  and thus can be approximated by subgradients using optimal dual

solutions (i.e. the recourse function also generates an optimality cut that is subsequently included in the master problem as an additional constraint). Each second-stage problem that is infeasible for the proposed first-stage decisions generates a feasibility cut (using the extreme rays of the dual polyhedron of the recourse problem). In the integer L-shaped method, after the traditional L-shaped method converges (in which we solve relaxed master problem), the binary requirements on  $\chi$  are reinforced. The integer optimality cuts along with traditional optimality cuts are also added to master problem and the procedure continues until a termination criterion is met.

Let us assume that the general form of integer master problem of BCSP is as follows:

$$\min 0 + \sum_{\omega \in S} \rho^\omega \theta_\omega \quad (4.46)$$

$$s.t. \quad \theta_\omega + E_{\omega,\delta} x \geq e_{\omega,\delta} \quad \forall \omega \in S, \forall \delta \in I \quad (4.47)$$

$$D_{\omega,\delta} x \geq d_{\omega,\delta} \quad \forall \omega \in S, \forall \delta \in I \quad (4.48)$$

$$Ax = b, x \in \{0, 1\} \quad (4.49)$$

and the general form of subproblems for each scenario  $\omega$  ( $\omega = 1, \dots, |S|$ ) is:

$$\min (q^s)^T y^\omega \quad (4.50)$$

$$s.t. \quad W y^\omega = h_\omega - T_\omega x^v \quad (4.51)$$

$$y^\omega \geq 0 \quad (4.52)$$

where  $x$  are the first-stage,  $y^\omega$  are the second-stage decision variables. The  $x^v$  are the parameters that are inherited from the first-stage. The indice  $\delta$  in the master problem is the iteration index and set of iterations is represented by  $I$ . We introduce a new variable  $\theta_\omega$  accounting for the approximation of  $\mathfrak{J}(\chi)$  in  $\chi$  for each subproblem  $\omega$ . Constraint (4.47) stands for general formulation of optimality cuts, while constraint (4.48) presents feasibility cuts. We now will clarify the complete integer L-shaped method.

#### 4.4.1 Integer L-shaped method

Following is the integer L-shaped algorithm which is applied for the mentioned problem.



Step 1:  $\delta = 0$ ,  $LB = -\infty$ ,  $UB = \infty$ . Let  $f_s, o_s$  be feasibility and optimality cuts of problem  $s$ , respectively. First we relax the integer variables in the master problem and obtain the relaxed master problem (RMP).

Step 2: Solve the master problem with all optimality and feasibility cuts from previous iterations. Let  $(\bar{y}; \bar{z}; \bar{r})$  be the optimal solution. In the first iteration ( $\delta = 0$ ), there should be no optimality and feasibility cuts. Denote the objective value of the master problem as  $\vartheta_\delta$ . If  $\vartheta_\delta \geq LB$ , set  $LB = \vartheta_\delta$ .

Step 3: Check whether all second-stage problems are feasible for  $(\bar{y}; \bar{z}; \bar{r})$  with realizations of  $a_i^\omega, p_{iq}^\omega$  for each  $\omega = 1, \dots, |S|$  by solving following LP. The objective (4.53) measures the amount by which the constraints are violated, it is a sum of the values assigned to  $v_l^+, v_l^-$  where  $m$  is the set of constraints for each subproblem. If all  $|S|$  scenario subproblems are feasible, then go to Step 4. Otherwise, for each infeasible LP subproblem, let us say  $s^{th}$  problem, we introduce a new feasibility cut.

$$\min \tilde{\psi}(v_l^+, v_l^-) \quad (4.53)$$

$$s.t. \quad W y^\omega + v_l^+ - v_l^- = h_\omega - T_\omega x^v \quad (4.54)$$

$$y^\omega, v_l^+, v_l^- \geq 0 \quad \forall l \in m \quad (4.55)$$

Assume that the dual vector of above-mentioned formulation is  $\pi^v$  (associated with the optimal basis in problem (4.50-4.52)). The following constraint (4.56) is the feasibility cut which is added to master problem for each scenario.

$$D_{f_s} x \geq d_{f_s} \quad (4.56)$$

$$D_{f_s} := (\pi^v)^T T_\omega \quad (4.57)$$

$$d_{f_s} := (\pi^v)^T h_\omega \quad (4.58)$$

Repeat Steps 2 and 3 until all the second-stage problems are feasible.

Step 4: Solve all the second-stage problems (4.50)-(4.52) to optimality. Now, we extract the optimal dual vector of constraints in (4.51) and call this vector  $\pi_1^\delta(\omega)$  where  $\delta$  stands for the iteration counter.

We then update the upper bound (UB) which is the weighted sum of the objectives of all scenario subproblems where  $\bar{y}^\omega$  is the optimal solution to decision variables for scenario  $\omega$ :

$$UB = \min\{UB, 0 + \sum_{\omega \in S} \rho^\omega(q^s)^T y^{\bar{\omega}}\}$$

In the traditional L-shaped, dual information from all of the subproblems are combined to generate a single optimality cut. In our algorithm, a multicut version, in which optimality cuts from each individual subproblem are added, is implemented (see an example You and Grossmann (2011)).

$$\theta_\omega + E_{o_s} x \geq e_{o_s} \quad \forall \omega \in S \quad (4.59)$$

$$E_{o_s} := \pi_1^\delta(\omega) T_w \quad (4.60)$$

$$e_{o_s} := \pi_1^\delta(\omega) h_w \quad (4.61)$$

If we are in the phase of solving the integer master problem, the following optimality cuts are also added to master problem. Given  $x^* \in \{0, 1\}$ , we let  $G(x^*) := \{i : x_i^* = 1\}$  for a master problem solution. The integer optimality cut at  $x^*$  is then defined as (4.62). This cut is proposed by Laporte and Louveaux (1993).

$$\sum_{\omega \in S} \theta_\omega \rho^\omega \geq (\mathfrak{J}(x^*) - LB) \left( \sum_{i \in G(x^*)} x_i - \sum_{i \notin G(x^*)} x_i - |G(x^*)| \right) + \mathfrak{J}(x^*) \quad (4.62)$$

The cut (4.62) puts a LB on the  $\theta_\omega$ . The proof of evidence for (4.62) can be found in Laporte and Louveaux (1993). We also formulate the no-good cut (Rei, Gendreau, and Soriano (2007)) which states that the variables which are in the set  $G(x^*)$  cannot take the value of one altogether in the optimal solution (4.63). This cut ensures that at least one of the binary variables will be changed in the value.

$$\sum_{i \in G(x^*)} x_i - \sum_{i \notin G(x^*)} x_i \leq |G(x^*)| - 1 \quad (4.63)$$

The no-good optimality cut (4.63) also guarantees that the solution generated in this iteration will be discarded from the feasible solution space.

Step 5: Check if  $(UB - LB)/UB \leq \epsilon$  ( $\epsilon$  is set to  $10^{-3}$ ). If this condition is not met and we are in the stage of solving RMP, obtain the optimality cut (4.59) and add it to the master problem,  $\delta = \delta + 1$ , and go to Step 2. If this condition is not met and we are in the stage of solving integer master problem, obtain the optimality cuts (4.59, 4.62, 4.63) and add them to the master problem,  $\delta = \delta + 1$ , and go to Step 2.

If the condition is met ( $(UB - LB)/UB \leq \epsilon$ ), check whether we solved RMP. If RMP is solved, impose the integrality of master problem variables back again,  $\delta = \delta + 1$ , set  $UB = \infty$  and go to Step 2. If it is the integer master problem which is solved, then Stop, extract the optimal solution.

#### 4.4.2 Accelerating strategies

To achieve faster convergence of the L-shaped solution approach for large-scale optimization problems, additional enhancement techniques are used. Computational complexity of master problem is one of main reasons of tardy convergence, this is due to weak cuts generated along the iterations. We suggest the enhancements of the two-phased solution method, the use of parallel computing, knapsack inequalities and combinatorial Benders' cuts. We now discuss each of these acceleration techniques.

The two-phased solution approach is about solving RMP first until L-shaped method converges. Since the convex hull of the feasible region for the original master problem is contained within the LP relaxation, all added cuts in the first stage are also valid for the integer master problem. Then integrality requirements are imposed in the second phase. Since it is easier to solve LP relaxation, such a solution method will accelerate the complete solution time (See Froyland, Maher, and Wu (2014) for a comparison between traditional and two-phased approach for a robust tail assignment problem).

The subproblem is an LP model that usually can be solved immediately. However, a high number of scenarios makes it time consuming to solve all of them in a single processor machine. What is more, in our settings, we assume that all scenarios are identically and independently distributed which means that scenarios do not have to share any information between them. For these reasons, we solve each of these subproblems concurrently in multi-thread environment by using OpenMP. The subproblems are assigned to each thread in numerical index order. Whenever all scenario subproblems are solved, exactly same cuts are generated. Maher, Desautniers, and Soumis (2014) compare single and multi-thread solution approaches for a two-stage stochastic programming problem and show the efficiency of parallelizing subproblem solvers.

We also formulate knapsack inequalities that use the information stored in the optimality cuts. At an iteration  $i$ , we know that  $UB_i \geq \sum_{\omega \in S} \theta_\omega \rho^\omega$ . What is more the optimality cuts generated for all scenario subproblems hold the property  $\sum_{\omega \in S} \theta_\omega \rho^\omega \geq \sum_{\omega \in S} \rho^\omega (e_{o_\omega} - E_{o_\omega} x)$ . This means that upper bound will be higher than RHS of the optimality cut which can be formulated in the form of (4.64). This cut can be added to master problem in the  $i+1^{th}$  iteration.

$$\sum_{\omega \in S} \rho^\omega (e_{o_\omega} - E_{o_\omega} x) \leq UB \quad (4.64)$$

Such inequalities of (4.64) have been suggested by Santoso, Ahmed, Goetschalckx, and Shapiro (2005). The authors note that solvers can generate useful cover inequalities which expedite the convergence.

In order to improve the performance of feasibility cuts in the integer master problem phase, we impose the combinatorial benders cuts (CBCs) similar to ones which are special versions of local branching technique of Codato and Fischetti (2006). In their paper, Codato and Fischetti (2006) argue that the contribution of CBCs for the case, in which objective function depends on both first and second stage variables, requires further computational investigation. In this study, we conduct this analysis. Since the objective function of the two-stage stochastic problem depends on the completion time of each vessel (second-stage) and QC assignment (first-stage) decisions, we have to use the continuous copies of  $y, z, r$  variables ( $\hat{z}_{ik}, \hat{r}_{iq}, \hat{y}_{ij}^{ss}, \hat{y}_{ij}^{sc}, \hat{y}_{ij}^{cs}, \hat{y}_{ij}^{cc}$ ) into the subproblems and generate the form of constraints required for CBCs as suggested by Codato and Fischetti (2006). We denote the row indices of constraints (4.27)-(4.32) by  $\Delta$  and the row indices for minimal infeasible subsystems (MIS) of the current subproblem by  $\Lambda$ . Note that the set of intersection ( $\Gamma$ ) of  $\Delta$  and  $\Lambda$  corresponds to row indice of MIS of each subproblem. The fundamental rationale of the CBC algorithm is that as long as  $\hat{z}_{ik}, \hat{r}_{iq}, \hat{y}_{ij}^{ss}, \hat{y}_{ij}^{sc}, \hat{y}_{ij}^{cs}, \hat{y}_{ij}^{cc}$  make subproblem infeasible, it indicates that at least one binary variable in  $y, z, r$  has to be changed to break the infeasibility. This statement can be translated to a linear inequality called the combinatorial Benders' cut in (4.65).

$$\begin{aligned} \sum_{t \in \Gamma_y, y^*(t)=0} y(t) + \sum_{t \in \Gamma_y, y^*(t)=1} (1-y(t)) + \sum_{t \in \Gamma_z, z^*(t)=0} z(t) + \sum_{t \in \Gamma_z, z^*(t)=1} (1-z(t)) \\ + \sum_{t \in \Gamma_r, r^*(t)=0} r(t) + \sum_{t \in \Gamma_r, r^*(t)=1} (1-r(t)) \geq 1 \end{aligned} \quad (4.65)$$

The cut (4.65) guarantees that at least one of the  $y, z, r$  variables which make the subproblem infeasible will be changed to its alternative value. This means that if, for example,  $z_{11}, r_{23}, r_{34}$  are the variables that cause the infeasibility in the subproblem, and they have the values  $z_{11} = 1, r_{23} = 1, r_{34} = 1$ . This cut guarantees that at least one these variables will be set to zero to break the infeasibility.

**Proposition:** The combinatorial Benders' cuts can be used as the feasibility cuts of the integer L-shaped method and they will not violate the exactness of the algorithm.

**Proof:** It is very clear that the feasibility problem of [RP] cannot be affected by the objective function, it is about the constraints (4.10)-(4.34). Then we can assume that the objective function is minimization of  $\sum_{i \in V} \hat{\mathbf{0}} \cdot \frac{c_i}{d_i} + \hat{\mathbf{0}} \cdot \sum_{i \in V} \sum_{q \in R_i} q r_{iq} p_{iq}^{\omega}$  for the feasibility problem. This property ensures conditions " $c = 0, d = 0$ " presented in Codato and Fischetti (2006) which proves the proposition  $\square$

### 4.4.3 Alternative strategies

The performances of Benders' decomposition inspired methods depend on various factors. Many researchers note that the tradeoff might also differ between formulations and instance sizes. For such reasons, we further investigate the decomposition alternatives that might perform well for the BCSP.

Trukhanov, Ntaimo, and Schaefer (2010) discuss that for integer L-shaped methods with every cut you add, RMP becomes just a bit harder, and when you solve the master problem with integrality restrictions, it becomes burdensome very quickly. In this sense, we test both single cut and multi cut options which simply affects the number of cuts added to master problem in every iteration.

Alternative-1 (Multicut approach): The details of multicut have been explained in the previous sections where the idea is to generate an optimality cut for each of the scenario subproblems.

Alternative-2 (Singlecut approach): The singlecut approach is performed in the same manner with the following modification to the multicut approach. After solving each master problem, instead of adding an optimality cut to the master problem for each subproblem of the form (4.59), we aggregate all generated

optimality cuts at that iteration into one single optimality cut reflected by (4.66).

$$\theta + E_{o_s}x \geq e_{o_s} \quad (4.66)$$

$$E_{o_s} := \sum_{\omega \in S} \rho^\omega \pi_1^\delta(\omega) T_w \quad (4.67)$$

$$e_{o_s} := \sum_{\omega \in S} \rho^\omega \pi_1^\delta(\omega) h_w \quad (4.68)$$

This single cut is formed by multiplying each generated optimality cut with the probability of corresponding scenario and summing them up. For singlecut version, the general form of master problem will also be changed. The objective (4.46) will be replaced by  $\min 0 + \theta$ , while constraint (4.47) will be reformulated as  $\theta + E_\delta x \geq e_\delta \quad \forall \delta \in I$ . Finally the integer optimality cut (4.62) should have a LHS of  $\theta$ .

## 4.5 Computational results

In this section, we present computational results of integer L-shaped methods and DEF on various test problems. All computational experiments are run on a 32 core AMD Opteron at 2.8Ghz and 132Gb of RAM computer. Runtimes are measured in seconds. The integer L-shaped approach is implemented in C++ using Concert Technology, and models are solved by using CPLEX 12.6.1. All available cores are used in parallel to solve the subproblems. The master problem is solved with the CPLEX option of emphasizing the optimality by using one thread. The preprocessor and dynamic cut generator options are disabled for solving the subproblems. The imposed computational time limit is 18000 seconds (5 hrs) for all models and L-shaped approaches.

### 4.5.1 Data set

The talks with various container terminals have shown us that the arrival time and exact container load (i.e. consequently processing time) of the upcoming vessels realize approximately two days before the expected arrival time. What is more, three days before of vessel's expected arrival, the shifts and QC maintenance plans are also available. This information supports the assumption of the simultaneous realization of arrival and processing time. Such a planning policy

results in a 2-3 days of planning horizon. The benchmark used in this study consists of 10, 26 ( $N$ ) vessels and we assume that 8 ( $Q$ ) QCs are available to load/unload the containers. We test the methods for three different scenario settings ranging between 10 to 640 scenarios for each vessel-QC combination. This means that in total, we generate 6 different data sets to test the approaches.

In order to generate each particular instance, we use Chen et al. (2012) data as the seed. The size of each vessel is randomly generated between 4 and 20 vessel-holds where each of them requires some QC-hours randomly distributed between 1 and 6 hours. The arrival, processing time and  $\tau(=1)$  are all measured in hours. The berthing positions of vessels and positions of QCs are measured in meters along the berth. We assume that each vessel  $i$  has an  $R_i$  set ranging between 1 to 5 QCs, while  $cost_2$  are 1 for all QCs. The scenario information is randomly generated for both vessel arrival and processing time. Assuming that  $|S|$  scenarios are used in a setting, each scenario  $\omega$  has a probability  $\rho^\omega$  of  $1/|S|$ .

### 4.5.2 Results

Table 4.3 summarizes the computational results for two versions of the integer L-shaped method and DEF. In this table, the column "#" indicates the instance ID, while columns  $N$ ,  $|S|$  show the number of vessels and scenarios, respectively. The columns named "Z" hold the best upper bounds obtained, while "LB" report the best LBs found within the time limit. The optimality gap ( $G = \frac{Z-LB}{Z}$ ) is calculated between the upper and lower bounds. The columns " $T_{OPT}$ " show the overall time spent to solve the problem. The integer L-shaped methods have two additional columns which present the performance of first-phase of integer L-shaped approach (where we solve RMP). The columns named " $Z_{RMP}$ " show the best bounds obtained when first-phase is converged, while " $T_{RMP}$ " report the time that it takes to converge for the first-phase.

**Table 4.3:** Computational results: Integer L-shaped method and DEF

			Integer L-shaped (Alternative-1)					
#	N	S	Z	LB	G(%)	T <sub>OPT</sub>	Z <sub>RMP</sub>	T <sub>RMP</sub>
1	10	10	3238.2	2541.3	21.5	18000	2258.0	87
2		320	3722.6	2933.9	22.4	18000	2584.2	2327
3		640	3986.9	2837.0	28.8	18000	2672.2	5216
4	26	10	12658.3	5401.4	57.3	18000	4644.1	1324
5		320	34841.6	19874.2	42.9	18000	16870.5	7533
6		640	38153.9	20145.3	47.2	18000	17545.6	11129
			Integer L-shaped (Alternative-2)					
1	10	10	3377.2	2474.3	26.7	18000	2258.0	187
2		320	3831.6	2927.9	23.5	18000	2584.2	2427
3		640	4138.3	2843.0	31.3	18000	2672.2	6018
4	26	10	12744.3	5516.2	56.7	18000	4644.1	1814
5		320	33612.8	20182.3	39.5	18000	16870.5	8612
6		640	37871.0	21548.6	43.1	18000	17545.6	12012
			DEF					
1	10	10	3128.9	3027.9	3.2	18000	-	-
2		320	3796.2	2642.0	30.3	18000	-	-
3		640	4159.8	2817.4	32.2	18000	-	-
4	26	10	9651.0	7218.9	25.2	18000	-	-
5		320	28021.8	19820.5	29.2	18000	-	-
6		640	31827.1	20814.9	34.6	18000	-	-

Results show that solving BCSP under uncertainty is a challenging problem. It is clear that increasing the number of vessels for the same amount of available QCs makes it even harder to solve. In average, the optimality gap obtained is 24.4% for 10 vessels instances, while this is 39.3% for 26 vessels instances.

For smaller number of vessels, integer L-shaped methods mostly obtain better upper and lower bounds compared to DEF. This comment is stronger for the increasing number of scenarios. Such an outcome supports the rationale of the use of an integer L-shaped method since this method solves each subproblem separately. For smaller number of vessels, integer L-shaped methods find better upper and lower bounds for 320, 640 scenario instances. However, the DEF outperforms the decomposition results for 10 scenario instances. For larger number of vessels, the problem becomes strongly complicated for both DEF and decomposition methods. Results show that lower bounds of integer L-shaped methods are mostly better than DEF for large scenario trees (320, 640 scenarios). The upper bounds by DEF are however better for all scenario sizes for large number of vessel instances. We can note that DEF is still competitive



with a decomposition method for this version of the problem. It should be noted that for instances with 26 vessels, the original version of valid inequality (4.38) contains more 70 million constraints, so this valid inequality is just generated with the assumption that at most 5 vessels can be in the berth at the same time for these instances. The exactness of the method will not be violated since this constraint is a valid inequality for the master problem.

The comparison between alternative integer L-shaped methods is also relevant. Results show that depending on the size of instance, the first-phases of integer L-shaped methods converge after reasonable amount of time. In the average, multicut spends less time (4602 sec) to converge compared to singlecut (5178 sec) in the first-phase. This is expected, because multicut version is proved to be time-efficient for solving relaxed master problems. The values of lower bound after the converge of the first-phase are the same for the two versions. After imposing the integrality restrictions in the second-phase, the mixed integer linear programming master problem exploits easily for larger problem sizes. Results show that single cut version which has fewer constraints compared to multicut version obtains better results when the number of vessels is increased. This comment is supported from the observation that the gaps obtained for 26 vessels instances are always better for singlecut version. For 10 vessels instances, multicut integer L-shaped method is still better for all scenario sizes. The results of the first-phase of integer L-shaped method are lower bounds for the complete problem. Hence we can analyze the improvements from the first-phase with respect to lower bounds. For 10 vessels instances, the integer L-shaped methods obtain an average of 8.6% improvement over the first-phase lower bounds, while this increase is 15.4% for 26 vessels instances.

We further investigate the contribution of the stochastic programming approach. Getting an average of all random scenario information and including them in one model for simplifying the solution is called the expected value (EV) problem. We first solve the DEF for only one scenario in which we use average scenario information for stochastic parameters (EV problem). The expectation of the expected value (EEV) is obtained by resolving RP for all scenarios, after fixing the first-stage values obtained from the EV solution. The weighted average of the optimal objective value for each scenario is the EEV. The value of stochastic solution (VSS) is the cost of disregarding arithmetic mean of vagueness while making decision. The VSS is calculated by subtracting the best upper bound of integer L-shaped or DEF from EEV (i.e.  $VSS = EEV - RP$ ).

Another indicator is about the cost of uncertainty or the maximum amount the decision maker is willing to pay in order to make a decision without uncertainty. In order to calculate this parameter, decision maker should be supplied with the solution of complete information. This is called as wait-and-see (WS) problem. In this problem, DEF is solved for each scenario separately, afterwards

expected value of all scenario subproblems are found with the weighted average of the optimal objective value for each scenario. The expected value of perfect information (EVPI) is the difference between solution of WS problem and the best upper bound of integer L-shaped or DEF (i.e.  $EVPI = RP - WS$ ). Higher EVPI means that uncertainty is important to the problem. The RP values are obtained from the best upper bounds in Table 4.3. We report the EEV, RP, WS VSS, EVPI results for each instance in Table 4.4.

**Table 4.4:** Importance of uncertainty for berth and QC scheduling problem

		EEV	RP	WS	VSS	EVPI	EVPI/RP (%)	VSS/RP (%)
N, S	10v,10s	3149.1	3128.9	3040.1	20.2	88.8	2.83	0.64
	10v,320s	3801.8	3722.6	2488.9	79.2	1233.7	33.14	2.12
	10v,640s	4041.3	3986.9	2912.6	54.4	1074.3	26.95	1.36
	26v,10s	9716.8	9651.0	6971.8	65.8	2679.2	27.76	0.68
	26v,320s	28435.1†	28021.8	††	413.3	-	-	1.47
	26v,640s	30582.6‡	31827.1	††	-1244.5	-	-	-

† represents that CPLEX had 1% gap after 5 hrs in order to solve EEV, while †† means that many instances of 320/640 scenarios cannot be solved to optimality in 5hrs, for this reason WS are not presented,

‡ represents that CPLEX had 4% gap after 5 hrs in order to solve EEV

Results show that the stochastic programming solutions payoff for almost all instances except the 26v,640s instance, this is mainly because very large scale instances cannot be solved perfectly with the current methods available to us. For 10 vessels instances, VSS values are all positive. This means that stochastic programming methods would obtain better results compared to expected value based methods. For 10 vessels instances, the EVPI remains a lot higher than VSS value. This means that further improvements can be achieved in the stochastic programming methods. High values of EVPI point out that management of uncertainty is very important for this problem. For 10 vessels instances, higher number of scenarios traditionally results in higher EVPI/RP(%) values. This can be observed from 10 v 320/640 scenarios. However, this is not the case between 320 v 640 comparison, because RP is not solved perfectly to optimality by our methods for these instances.

For 26 vessels instances, results are more erratic. Due to computational complexity of the problems for larger scenario sizes, we cannot solve many problems to optimality to obtain EEV and WS, what is more RP solutions presented have high optimality gaps. For these reasons, larger scenario sizes might be misleading to evaluate. However, results show that VSS values for 10, 320 scenarios are positive. This supports that an improvement has been made compared to expected value problem. For the largest instance, solving EEV obtains a better solution compared to stochastic programming method. This is mainly because

even RP result has an optimality gap of 34.6%.

Results show that uncertainty is an important part of decision making for BCSP. Having high uncertainties, terminals might take risk-averse approaches to solve this problem. In the next subsection, we will analyze the results of our formulations for risk-attributed settings.

### 4.5.3 Risk-averse stochastic programming for BCSP under uncertainty

The traditional two-stage stochastic programming which is based on expected value of second stage problem has a risk-neutral approach. Since the influence of berth and QC scheduling problem has a strong impact on the terminal performance, some terminals might select a risk-averse approach. Then they might plan the seaside operations with the safe mode. In this section, we use the conditional-value-at-risk (CVaR) as the risk measure for the BCSP and it is incorporated into the objective function.  $CVaR_\alpha$  is the conditional expected value exceeding the value-at-risk at the confidence level  $\alpha$ .

Noyan (2012) has proved that for the case of finite probability space, one can formulate a general mean-risk problem to solve two-stage stochastic programming formulations. In this approach,  $\lambda$  is used as the positive trade-off coefficient representing the exchange rate of mean cost for risk (i.e. the risk coefficient). We will test different values of the risk coefficient (Any terminal may select a level of risk coefficient for its attitude against risk).

Computational results show that DEF is competitive with the integer L-shaped methods for BCSP. For this reason, we will present the DEF results for the mean-risk [MR] stochastic programming problem for BCSP. As we have mentioned decision variables for the DEF in section 4.4, we will not introduce them into detail. Following model is used as the DEF:

$$[MR] = \min \sum_{\omega \in S} \rho^\omega \left( \sum_{i \in V} cost^1_i \frac{C_{i\omega}}{d_i} + cost^2 \sum_{i \in V} \sum_{q \in R_i} qr_{iq} p_{iq}^\omega \right) + \lambda \left( \eta + \frac{1}{1 - \alpha} \sum_{\omega \in S} \rho^\omega v_\omega \right) \quad (4.69)$$

subject to

$$(4.2) - (4.6) \quad (4.70)$$

$$s_{i\omega} \geq a_i^\omega \quad \forall i \in V, \omega \in S \quad (4.71)$$

$$c_{i\omega} \geq s_{i\omega} + \sum_{q \in R_i} p_{iq}^\omega r_{iq} \quad \forall i \in V, \omega \in S \quad (4.72)$$

$$s_{j\omega} \geq s_{i\omega} + M_1(y_{ij}^{ss} - 1) \quad \forall (i, j) \in \mathbb{U}, \omega \in S \quad (4.73)$$

$$c_{j\omega} \geq s_{i\omega} + M_1(y_{ij}^{sc} - 1) \quad \forall (i, j) \in \mathbb{U}, \omega \in S \quad (4.74)$$

$$s_{j\omega} \geq c_{i\omega} + M_1(y_{ij}^{cs} - 1) \quad \forall (i, j) \in \mathbb{U}, \omega \in S \quad (4.75)$$

$$c_{j\omega} \geq c_{i\omega} + M_1(y_{ij}^{cc} - 1) \quad \forall (i, j) \in \mathbb{U}, \omega \in S \quad (4.76)$$

$$xs_{ki\omega} \geq l_i + M_2(z_{ik} - 1) \quad \forall i \in V, k \in \Phi, \omega \in S \quad (4.77)$$

$$xs_{ki\omega} \leq l_i + M_2(1 - z_{ik}) \quad \forall i \in V, k \in \Phi, \omega \in S \quad (4.78)$$

$$xc_{ki\omega} \geq l_i + M_2(z_{ik} - 1) \quad \forall i \in V, k \in \Phi, \omega \in S \quad (4.79)$$

$$xc_{ki\omega} \leq l_i + M_2(1 - z_{ik}) \quad \forall i \in V, k \in \Phi, \omega \in S \quad (4.80)$$

$$xs_{kj\omega} \leq l_i + M_2(3 - z_{ik} - y_{ij}^{ss} - y_{ji}^{sc}) \quad \forall (i, j) \in \mathbb{U}, k \in \Phi, \omega \in S \quad (4.81)$$

$$xs_{kj\omega} \geq l_i + M_2(z_{ik} + y_{ij}^{ss} + y_{ji}^{sc} - 3) \quad \forall (i, j) \in \mathbb{U}, k \in \Phi, \omega \in S \quad (4.82)$$

$$xc_{kj\omega} \leq l_i + M_2(3 - z_{ik} - y_{ij}^{sc} - y_{ji}^{cc}) \quad \forall (i, j) \in \mathbb{U}, k \in \Phi, \omega \in S \quad (4.83)$$

$$xc_{kj\omega} \geq l_i + M_2(z_{ik} + y_{ij}^{sc} + y_{ji}^{cc} - 3) \quad \forall (i, j) \in \mathbb{U}, k \in \Phi, \omega \in S \quad (4.84)$$

$$xs_{ki\omega} - xs_{k-1,i\omega} \geq 0 \quad \forall i \in V, k \in \Phi \setminus \{1\}, \omega \in S \quad (4.85)$$

$$xc_{ki\omega} - xc_{k-1,i\omega} \geq 0 \quad \forall i \in V, k \in \Phi \setminus \{1\}, \omega \in S \quad (4.86)$$

$$s_{j\omega} - c_{i\omega} \geq \tau \quad \forall (i, j) \in \mathbb{T}, \omega \in S \quad (4.87)$$

$$(4.35) - (4.45) \quad (4.88)$$

$$v_\omega + \eta \geq \sum_{i \in V} (n_i^1 \frac{c_{i\omega}}{d_i} + n^2 \sum_{q \in R_i} q r_{iq} p_{iq}^\omega) \quad \forall \omega \in S \quad (4.89)$$

$$(4.7) \quad (4.90)$$

$$\eta \in \mathbb{R}, s_{i\omega}, c_{i\omega}, v_\omega \in \mathbb{R}^+ \quad \forall i \in V, \omega \in S \quad (4.91)$$

$$xs_{ki\omega}, xs_{ki\omega} \in \mathbb{R}^+ \quad \forall i \in V, k \in \Phi, \omega \in S \quad (4.92)$$

Constraint 4.89 has  $n_i^1, n^2$  parameters which are  $cost_i^1, cost^2$ , written so due to space limit.

There are two new decision variables for the mean-risk problem. Note that the

**Table 4.5:** The expected cost vs risk-measure for different risk-averseness levels

N, S : 26v,320s	$\lambda=0, \alpha=0.9$	$\lambda=0, \alpha=0.7$	$\lambda=5, \alpha=0.9$	$\lambda=5, \alpha=0.7$
RP	28021.8	28021.8	28192.6	28107.3
CVaR $_{\alpha}$	280218.3	93406.1	47257.8	40552.9
MR	28021.8	28021.8	264481.6	230871.8
G(%)	29.2%	29.2%	26.4%	28.8%
N, S : 26v,320s	$\lambda=10, \alpha=0.9$	$\lambda=10, \alpha=0.7$	$\lambda=100, \alpha=0.9$	-
RP	28370.6	29424.8	28364.3	-
CVaR $_{\alpha}$	48907.1	41530.3	48305.1	-
MR	517441.3	444727.3	4858874.4	-
G(%)	28.3%	30.8%	27.1%	-

variable  $\eta$  is a first-stage variable which retrieves an approximation of the value-at-risk and the excess variables,  $v_{\omega}$ ,  $\omega \in S$  are second-stage variables. Objective function (4.69) is a combination of the expected value of the cost function and perception of risk attributed to scenarios. Note that increasing the value of  $\lambda$  would increase the relative importance of the risk term and so would lead to more risk-averse policies. The specified  $\alpha$  level represents the risk preference in percentage terms, it quantifies the mean value of the worst  $(1-\alpha)100\%$  of the total costs. When  $\alpha$  increases the corresponding value-at-risk increases and CVaR $_{\alpha}$  (i.e.  $\eta + \frac{1}{1-\alpha} \sum_{\omega \in S} \rho^{\omega} v_{\omega}$ ) accounts for the risk of larger realizations. Thus, larger  $\alpha$  values would also lead to more conservative policies, which give more weight to worse scenarios. Constraint (4.89) sets the excess variables for each scenario which the difference between first stage variable  $\eta$  and the expected cost value.

In order to test the DEF for the mean-risk problem, we focus on 26 vessels, 320 scenarios instance. In Table 4.5, we present the value of expected cost for different risk-attribute settings. Results show that for extreme risk-averse scenarios the planning is mainly done for the case of worst-scenario outcomes. For this reason, the expected cost value increases as the decision leans more on conservative policies.

Results show that larger  $\lambda$  values provide higher expected cost for the similar optimality gaps. This means that these plans are formed according to worst case scenarios (very late arrival or longer processing times) for larger  $\lambda$  values. However, we cannot make such a claim for  $\alpha$ , i.e. larger  $\alpha$  values do not increase the expected cost all the time. For a fixed  $\lambda$  value, the expected cost might increase or decrease according to results. This specific comparison might be misleading since the gap of DEF is different for two cases ( $\lambda=5, \alpha=0.9/0.7$ ). Regarding the value-at-risk for similar optimality gap values, there is a clear

decrease in the value with the increasing  $\lambda$  values. This is expected because putting a higher risk-averseness will result in lower risk measure factor. What is more, for larger  $\alpha$  values, we obtain higher the value-at-risk value by its definition.

## 4.6 Conclusion and future research

In this chapter, we have presented a novel two-stage stochastic programming approach to solve berth and quay crane scheduling problem under arrival time and processing time uncertainties. The studies in container terminal operations mostly rely on estimated deterministic data. We go beyond these assumptions and use discrete probability distributions to reflect scenarios for stochastic parameters. The stochastic programming model that we suggested is also solved with an integer L-shaped method. We also present a number of valid inequalities to enhance the formulations. The expected two-three days of planning horizon makes the problem more applicable for terminal operators. Computational results show that container terminals can make measurable cost savings by implementing stochastic programming methods rather expected value based solution methods. We also show that for higher number of scenarios, the integer L-shaped method outperforms DEF both for upper and lower bounds.

We also conduct an analysis to evaluate different versions of the integer L-shaped method in which the number of optimality cuts generated in each iteration differs. Results point out that contrary to traditional L-shaped method to solve LP master problems, in integer L-shaped method, singlecut approach outperforms multicut for larger scenario trees and larger instance sizes. The higher number of constraints added to master problem in multicut version makes it harder to solve when integrality of binary variables are imposed. We evaluate this explanation as the reason of degrading performance of multicut version against single cut for enlarging problem sizes.

We also formulate a risk-attributed stochastic programming model for the problem. Results for fairly large scale instance show that putting a high risk-averse profile increases the overall costs, adverse it reduces the risk of not finishing the operations too late.

Future research should be focused on improving the performance of integer L-shaped method. Some properties of subproblem result in weaker cuts, we intend to come up with alternating optimality cuts for the mixed integer master problem. What is more, a branch-and-cut based integer L-shaped can be implemented. In this method, we are not expected to solve the master problem to

optimality in each iteration, instead we can branch on the current node which will reduce the computation effort to explore branch-and-bound tree at every iteration. Regarding the problem definition, we can incorporate the unexpected QC breakdowns into the two-stage stochastic formulation. A possible breakdown of a QC will not only increase the processing time of a vessel, it will also affect the movement of all QCs around it. We can integrate this property into the problem definition.

## CHAPTER 5

# A survey on the Ship Loading Problem

---

Recent statistics show that large container terminals can process more than 30 million containers a year, and are constantly in search for the better ways to optimize processing time, deliver high quality and profitable services. Some of the terminal decisions are, however, dependent on externalities. One of those is the ship loading process. Based on the stowage plan received by liner shippers, terminal operators plan the execution of load and discharge operations. In this chapter we present a literature review for the Ship Loading Problem, where stowage and loading sequencing and scheduling are integrated to improve the efficiency of the ship handling operations. We present a survey of the state-of-the-art methods and of the available benchmarking data.<sup>1</sup>

## 5.1 Introduction

The World economy has always relied on the ability of transporting goods. With the introduction of containerized shipping, global supply chains have flourished

---

<sup>1</sup>C. Iris, D. Pacino, "A survey on the Ship Loading Problem", *Computational Logistics, Lecture Notes in Computer Science*, Vol. 9335, pp. 238-251, published



and are now demanding more cost efficient and reliable transport. Liner shippers have responded by increasing the capacity of their fleet, deploying vessels of over 18.000 TEUs. Capacity is, however, not enough. A reliable service requires the goods to arrive on time, and it is here that container terminals play a major role. Recent statistics show that container terminal throughput, totaling worldwide 651.1 million TEUs in 2013 UNCTAD (2015), is estimated to increase by 5.6%.

As it can be seen from recent literature surveys Steenken et al. (2004); Bierwirth and Meisel (2010); Kim and Lee (2015), there is an increased interest on the use of optimization techniques for the planning of terminal operations. Moreover, there is a growing trend on integration approaches trying to increase the flexibility of the currently rigid hierarchical planning practices.

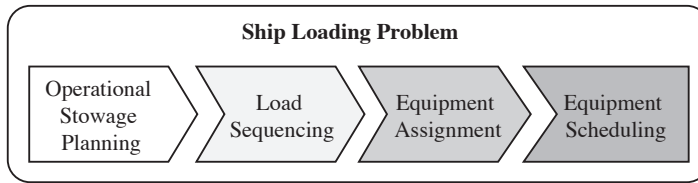
With this study, we aim at bringing the readers' attention to the Ship Loading Problem (SLP), where integration efforts go beyond terminal operations and try to reach the liner shipper as well. Traditionally, the liner shipper is responsible for generating stowage plans suitable for the current and future ports. Stowage coordinators spend great effort in generating plans that are both feasible in terms of vessel stability and efficient in terms of load and discharge operations. Terminals then plan the container sequencing accordingly. The position of a container in the vessel can, however, have a large impact on the needed transportation time of the container to and from the yard. More control over stowage planning would enhance the terminal's ability to efficiently plan ship handling operations. Common ground for both liner shippers and terminal operators, is the class based stowage plan. Class based stowage plans assign classes of containers to positions in the vessel rather than specific containers. Since the assignment of specific containers to each class has no impact for the objective of the liner shipper Pacino et al. (2011), this then leaves terminal operators with two planning decisions: 1) generating an operative stowage plan (a detailed stowage plan assigning containers to classes Monaco et al. (2014)) 2) sequencing of container load operations. Container sequencing is governed by precedences dictated by the physical position of the containers on the vessel e.g. the load sequence between two containers in the same stack (or row) cannot be changed, while it can if the containers belong to two different stacks. The integration between these two planning decisions allows a degree of freedom that, according to existing literature (e.g. Monaco et al. (2014); Steenken, Winter, and Zimmermann (2001)), has a large impact on terminal costs and handling time.

In this chapter we present a literature study of the SLP and its variants. We wish to illustrate the current state-of-the-art methods and identify interesting research opportunities. Also, we survey the currently used benchmarks and point out missing features in the conclusion. This chapter is organized as follows: First, the SLP is introduced in Section 5.2. The literature and benchmark review are presented in Section 5.3, including a comprehensive comparison table.

Finally, Section 5.4 draws conclusion and presents future work.

## 5.2 The Ship Loading Problem

When a container vessel arrives at port, handling equipment is immediately mobilized to service the ship. The management of loading operations, planning of the equipments to use and their scheduling is what we define as the SLP. In this sense, equipment scheduling heavily relies on the stowage plan of the vessel and on the sequence in which the containers are to be loaded. As depicted in Figure 5.1, we define the SLP as the integration of four terminal planning problems: operational stowage planning, load sequencing, equipment assignment and equipment scheduling.



**Figure 5.1:** Ship Loading Problem composition

Let us then define the SLP by describing each of those problems and their interaction. The term *operational stowage plan* was first introduced in Monaco et al. (2014) to distinguish between the stowage planning problem (solved by the liner shipper) and the operational refinement of it done by the terminal. Briefly, during stowage planning, an assignment of containers to vessel positions is performed. The assignment must fulfill stability requirement for its entire journey (not only the current port), while minimizing overstorage (re-handling of containers) and handling time. A class based stowage plan, is a plan where container types are assigned to vessel positions rather than actual containers, thus leaving the final match between containers and container classes to the terminal. We refer to this last container assignment as the *operational stowage planning problem*. The advantage of letting the terminal perform this operation can be easily described. Since the liner shipper has no knowledge of the yard arrangement, a detailed stowage plan might be costly in terms of transportation time. In Figure 5.2, it is shown how transportation time can be reduced by switching the assignment of two containers ( $c_1, c_2$ ) that belong to the same class ( $t_1$ ). Container classes are defined by their weight, length, height, etc. We refer the reader to Pacino et al. (2011) for a more in-depth description.

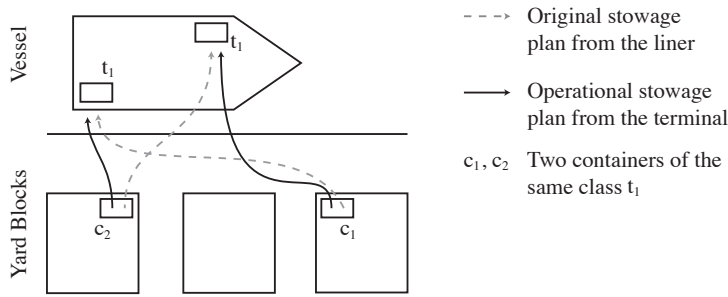


Figure 5.2: Operational stowage planning

Independently on whether the terminal receives a detailed or class based stowage plan, there is still a degree of flexibility, the optimization of the loading sequence. The *Load Sequencing* problem aims at better utilizing the yard equipment during vessel handling. The sequence in which containers must be loaded is governed by physical rules, e.g. the sequence between two containers destined to the same row cannot be changed. It is, however, up to the terminal to decide the sequence of containers on different rows or bays. Some terminal use predefined loading policies, such as "sea to land" where containers are loaded row or tier wise for the sea to the land side. This decision can also be seen as an optimization problem. Figure 5.3 shows an example where an optimized load sequence can greatly improve the total handling time (the value in parenthesis). Once a loading sequence is finalized, further handling time improvements can be achieved by allowing the pre-marshaling (Caserta and Voß (2009)) of yard blocks. This is, however, outside the scope of the SLP.

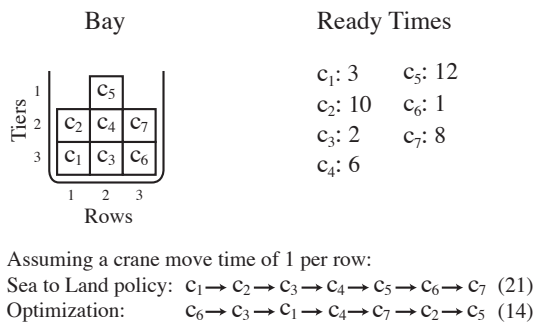


Figure 5.3: Load sequencing example.

Load sequencing, can only be done given the ready times of each container. By ready time we mean the time a container is ready to be loaded on the vessel, that being by a Quay Crane (QC) or reach stacker. The ready times depend on the equipment assigned to the vessel (QCs, straddle carriers, reach stackers, trucks etc.). At the same time the load sequence influences the scheduling of the equipment, thus making the integration of load sequencing and *equipment assignment and scheduling* an obvious choice.

Each of the previously mentioned planning problems defines the SLP, no matter whether they are solved hierarchically or as an integrated problem. The main objective of the SLP is the minimization of the total handling time. This can be interpreted as a minimization of re-handles in the yard and of the transportation times. Secondary objectives can be the minimization of costs associated to the used equipment. Hard constraints are mainly related to the stability requirements of the vessels and to the capacity of the terminal equipment.

## 5.3 Literature Review

The SLP is not well studied in the literature. The variety of settings, assumptions, and objectives considered in previous studies highlights the lack of a commonly accepted view of the problem. As mentioned in the previous section, we define the SLP as a combination of the operational stowage planning problem, the load sequencing problem and the equipment assignment and scheduling problem. Most of the works present in the literature can be classified as belonging to one of those sub problems, yet some authors present some integration efforts. The information needed to classify a problem has been inferred from the context, if not explicitly provided by the authors.

During our literature review we compare contributions based on three main aspects: problem structure, objective function and solution approach. The problem structure identifies the output of the planning problem, the constraints and assumption wrt. yard equipment, use of loading policies etc. Table 5.1 presents a comparison of the relevant literature. It is worth noticing that the number of scientific works is not overwhelming. From now on, we will assume the terminal as the decision maker and that only loading operations are taken into account.

### 5.3.1 Operative Stowage Planning Literature

To the best of the authors' knowledge, Monaco et al. (2014) is the only work that focuses entirely on the operative stowage planning problem. In Monaco et al. (2014) the input of the problem is a class-based stowage plan. The authors propose a mathematical formulation for the assignment of containers to classes. Since vessel stability constraints are already fulfilled in the class-based stowage plan, only stack weight capacity and weight sorting requirements are modelled. The model aims at minimizing the total travel distance and the number of re-handles in the yard. It is assumed that the terminal is operated by straddle carriers, and that there are no restrictions on the number of available vehicles. Since the mathematical formulation does not scale to realistic instances, a two-phase Tabu Search algorithm is proposed. A comparison with container terminal data, reveals that the model underestimates the yard re-handles, which the authors attribute to the stochastic nature of the problem.

#### 5.3.1.1 Stowage Planning with Terminal Considerations

In our definition of the SLP, we assume that stowage planning is the responsibility of the liner shipper. A number of works in the literature, however, do not share the same idea (e.g. Ambrosino and Sciomachen (2003); Sciomachen and Tanfani (2007); Zhao, Mi, Mi, and Chai (2013)). In these works, it is assumed that the terminal has full control over the positioning of containers in the vessel. The optimization problem that needs to be solved is then the *stowage planning problem*. Since it is not the scope of this survey to review stowage planning literature, we limit ourselves to the description of those works that include aspects of terminal optimization. It is also debatable (at least for liner shipping), whether the terminal should be responsible for the stowage plan.

The work presented in Imai, Nishimura, Papadimitriou, and Sasaki (2002) is one of the first stowage planning models that considers the minimization of yard-re-handles. The model formulates the stability of the vessel only in terms of GM (the distance between the center of gravity and the metacenter). No distinction is made between the different container types or their destinations. An estimate of the number of re-handles is calculated and included in the objective function as well. The estimate is shown to be a fair estimation of the real re-handles. In a later paper Imai et al. (2006), the authors include trim and heeling (longitudinal and transversal listing of the vessel) to the model. The stability constraints are still handled as an objective rather than a constraint. Also, the improved model includes the modeling of a yard with multiple rows. Due to the model complexity, the authors propose a solution approach based on genetic algorithms. As in

Monaco et al. (2014), there are no restrictions in terms of handling equipment.

In Ambrosino and Sciomachen (2003), a stowage planning model is used to analyze the impact of two different load policies: pre-marshaling (Caserta and Vofsi (2009)), and the sort and store policy. The model distinguished between container length, weight, discharge port and power requirements. Vessel stability is heuristically handled by balancing the front-back and right-left side of the vessel (an in-depth description of the model can be found in Ambrosino et al. (2004)). The generated stowage plan is then evaluated in terms of yard re-handles using the two policies. The stowage planning model was also used for the implementation of the sort and store procedure, which resembles just-in-time planning where the stowage plan is repetitively computed for the current subset of available containers. The authors argue that equipment costs must be taken into consideration before a clear conclusion can be drawn.

Another terminal efficiency analysis, that uses a stowage planning model, is presented in Sciomachen and Tanfani (2007). Here the same stowage planning problem as in Ambrosino et al. (2004) is considered. The problem is solved using a 3D-Bin Packing approach that, included into a hierarchical heuristic procedure, models the assignment of containers to QC. In a first phase the set of containers is heuristically distributed among the cranes. The containers are then assigned to the vessel by the stowage planning procedure. Mind that here, as an extra requirement, a stowage planner has identified subsections of the vessels to be used for containers with specific discharge ports. The generated plan is then evaluated in terms of quay crane productivity. A simplified version of this problem is studied in Zhao et al. (2013) where the authors wish to minimize the concentration of containers coming from the same yard block and thus focus on decreasing the interference in the yard. A mathematical model is presented, but no comments are given to the effectiveness of the approach in terms of terminal operations.

### 5.3.2 Load Sequencing Problem Literature

Literature on the Load Sequencing Problem is scarce. To the best of the authors' knowledge, only five works focus on this problem, yet the approach to the problem is very different. Differently than all the works surveyed until now, Meisel and Wichmann (2010) approaches the minimization of containers reshuffles directly within the vessel. The authors argue that the best loading sequence can be obtained by allowing changes to the stowage plan of each bay. The problem deals with finding a sequence of container moves that converts a given arrival configuration of a bay into a detailed stowage plan configuration within minimum service time. The objective is defined as the total processing time

of container moves and empty crane movement. This definition of the problem enables the exploitation of quay crane double cycling (i.e., alternating loading and unloading operations), and minimizes internal reshuffles within bays. The approach assumes that a departure configuration is given for the vessel, then no considerations are made to the impact of the stowage changes on the next port of calls. The proposed model is solved using a Greedy Randomized Adaptive Search Procedure (GRASP).

The following works move the focus on yard operations. In Lee, Kang, Ryu, and Kim (2005) the aim is the generation of balanced QC loading plans with respect to Transfer Crane (TC) assignment. The objective is the minimization of the total travel distance between all TCs and their setup cost when moved to pick up containers in a different yard block. The authors propose the following approach to cope with the complexity of the problem. Given a detailed stowage plan, containers are grouped into classes (by size and destination, effectively generating a class-based stowage plan). Then, in a hierarchical manner, the load sequencing is performed. Since containers of the same class are distributed among different yard bays, the method first performs an overall sequencing deciding how many containers should be moved from which yard-bay to which vessel bay. In a second stage, the detailed sequencing of each container within the container groups is calculated. The second stage problem is solved using the beam search heuristic proposed in Kim et al. (2004), while for the first stage, the authors propose an ant colony optimization, a Tabu search and a hybrid of these methods.

In Bian, Shao, and Jin (2015), the decision is to determine the loading sequence of containers with the aim of minimizing the number of re-handles. It is assumed that a detailed stowage plan is given, that only one QC is available, and that the relocation of containers within a yard block is allowed. The authors have proposed a two-stage algorithm. In the first stage, a heuristic is developed to load the containers which do not need any relocations. In the second phase, a dynamic programming algorithm with heuristic rules is presented to solve the load sequencing problem for all of the remaining containers. The results are compared with an alternative re-handling strategy, where load containers are chosen from stacks with the smallest number of blocked containers. Re-handles are then reassigned to random stacks. The number of relocations is shown to be reduced by 46.5% compared to the alternative strategy.

Another work on the load sequencing problem is presented in Ji, Guo, Zhu, and Yang (2015). Differently than Bian et al. (2015), the authors consider a scenario with multiple QCs, and test three different container re-handling location strategies: nearest-stack, lowest-stack, and a mixture of those. Firstly, a mathematical model of the problem is formulated. A genetic algorithm is then presented and tested for two versions. One version assumes that a certain

loading sequence is given, while the other also determines the loading sequence along with the re-handling strategy within a yard block. The results show that the number of re-handles is reduced up to 30% compared to state-of-the art solutions which assumes a given load sequencing strategy.

A different approach is taken in Legato and Mazza (2013), where the focus is on the calculation of the exact number of re-handles for a given load sequencing policy. They do this by simulating the picking strategies with a discrete-event based simulation model. The simulation model considers the availability of straddle carriers, the number of reshuffles to reach a stacked container, and the availability of buffer space under the crane. Tests on the impact of multiple QCs on the operational efficiency of the loading plan are performed. The authors argue that the turnaround time of vessel loading is significantly reduced when assigning more straddle carriers and note that the number of re-handles does not change (see also Legato, Mazza, and Trunfio (2010) for a simulation-based optimization approach for both loading and unloading operations).

### 5.3.3 Equipment Assignment and Scheduling Literature

Many works in the literature touch on different aspects of equipment management. Some examples are the determination of the number of the yard equipment to use Vis et al. (2005), the assignment of equipment to QCs and/or containers Lee, Chew, Tan, and Wang (2010b); Grunow, Guenther, and Lehmann (2006); Nishimura et al. (2005); Bish et al. (2005); Kim and Kim (1999c); Jung and Kim (2006), and sequencing and scheduling of the yard equipment Bose et al. (2000); Lee, Cao, and Meng (2007); Lau and Zhao (2008); Kim and Kim (2003); Bish et al. (2005); Li and Vairaktarakis (2004); Kim and Kim (1999c); Jung and Kim (2006); Cao et al. (2010b).

Each of these problems has been studied extensively in the literature. For this survey, we decided to concentrate on two of the most popular works that incorporate the all above mentioned equipment management problems, and that are concentrated around container loading.

The focus in Kim and Kim (1999b) is the optimal routing of one TC in a container yard during loading operations. They decide the number of containers that a TC picks up at each yard-bay and they also determine the sequence of yard-bays that a TC visits during the loading of the vessel. It is assumed that QC work schedule is given, which defines how many containers of a specific type (size and destination) should be loaded in which vessel-bay at what time interval. The objective of the presented mathematical model minimizes the total container handling time, which consists of the setup time at each yard-bay and



the travel time between yard-bays.

The work presented in Bish et al. (2005) looks at the vehicle dispatching problem, which also assumes a given QC work schedule. The authors assume that a fixed number of vehicles has been assigned to serve each QC. The problem aims at minimizing the ship berthing time. They propose a greedy heuristic to solve the problem. For the single-crane case, they prove that the greedy algorithm is optimal. This does not hold for the multiple crane case. For multiple QC case, they provide a modification of the greedy algorithm which, compared to the results obtained with a mathematical model, finds better solutions than the original algorithm.

QCs are also considered as quay-side equipment which are attributed to the ship loading. In the most of the SLP studies, a QC working plan is given as input to the problem Kim et al. (2004); Kim and Kim (1999c); Jung and Kim (2006). This plan holds the QC split which points out which specific QC will work on each bay. Once a QC working plan is given, the QC scheduling for bay areas (see Meisel and Bierwirth (2011)) can be generated. The QC scheduling problem is, however, outside the scope of the SLP.

### 5.3.4 Integration Efforts

The literature on integration efforts for the SLP is also rather limited. To the best of the authors' knowledge, only one of the surveyed works actively models and optimizes the container sequencing. The other integration efforts focus on the combination of operational stowage planning and equipment planning, where a loading policy is assumed (e.g. sea-land, fill each stack first. etc.), leaving no space for the sequence optimization.

The pioneering paper on SLP is Steenken et al. (2001), which focuses on the generation of operative stowage plans and the allocation/schedule of straddle carriers. They assume that a given fixed number of straddle carriers is available for each QC. The problem described in Steenken et al. (2001) relies on the assumption that only one QC operates and disregards the equilibrium constraints of the vessel in the detailed stowage plan. A just-in-time scheduling model is solved when a group of container is ready to be retrieved from the yard. This model assigns each container to a specific slot and a straddle carrier. Later, in the same paper, the authors extend the approach to include multiple QCs. This is done by first solving the crane split problem and then applying the single crane heuristic to each QC. Authors assume two loading strategies: column-wise, where each stack is filled in sequence, or layer-by-layer. The objective is the minimization of lateness of the QC moves and the transportation time

between the yard and the quay area. To solve the problem, the authors present a mathematical model and a best-fit heuristic which can be applied for all QCs in parallel.

Another integration approach is Kim et al. (2004). Given the QCs schedule, this work combines operational stowage planning, load sequencing and TC scheduling, making this, to the best of the authors knowledge, the most complete integration effort. With respect to operational stowage planning, the presented non-linear mathematical model incorporates vessel stability considerations by imposing weight and height limit constraints on the stacks of the vessel. The sequencing is part of the model decisions, however, a column-wise policy is encoded into the objective. Only the travel time costs of the TCs can force the sequence away from this policy. The actual schedule for the loading of the containers is also modelled, aiming at minimizing reshuffles, TCs travel times and interferences. A two stage approach is used to solve the problem. The first stage sequences yard-clusters (like the first stage of Lee et al. (2005)), while the second sequences individual containers. Both stages are solved using beam search. In a later paper Jung and Kim (2006), the authors propose a genetic algorithm and a simulated annealing heuristic for solving the first stage problem. Although this problem is similar to Lee et al. (2005), we include it in this section since the scheduling of Yard Cranes (YCs) is also included. Here the objective is the minimization of the makespan for the YCs.

In Alvarez (2006) we find an integration approach that combines detailed yard equipment planning and scheduling, with operative stowage planning. A terminal operated by reach-stackers is considered. This work is particularly interesting for its presentation of a full mathematical model that includes all the aspects regarding reach-stackers routing and operations at the yard for each container. It is assumed that a selection is made between column-wise or layer-by-layer loading policy. Moreover it assumes that containers are expected to be loaded sea-to-land. As in previous approaches, the use of loading policies leaves no space for sequencing optimization. The model minimizes the number of re-handling, the movement of reach-stacker (i.e. distance traveled on the ground by the transport vehicles) and vessel instability. A solution of the model establishes which sections of the yard provide the containers of the needed type (with given numbers of containers) and generates a feasible tour of the yard to pick them up. A Tabu Search algorithm is described for solving the problem. A Lagrangian relaxation approach is proposed by the same author in a later work Alvarez (2008). A summary of the surveyed literature can be found in Table 5.1. The reader can compare the different manuscripts in terms of problem structure, objective function and solution approach.

Table 5.1: Ship loading problem literature abstract

		Problem Structure														Objective Function						Solution Approach															
		A				B		C		D			E			F		G		H		Type	$T_w$	$T$	$T^{Y-Q}$	$S_n$	$E_d$	$Q_b$	X	Y	Z						
Year	Authors	1	2	3	4	1	2	1	2	1	2	3	1	2	3	1	2	3	1	2	1	Min			*	*			1	1	1	2	3	4	5	6	7
2014	Moraho et al. (2014)	*						*							*						1	Min				*							*	*	*	*	
2003	Ambrosio and Schmecken (2003)		*			*		*		*		*		*							*	Min			*		*		*				*	*	*	*	
2006	Huai et al. (2006)											*		*							*	Min				*		*		*			*	*	*	*	
2007	Scamachen and Tantai (2007)					*		*		*		*		*					*		*	Min	*				*		*		*			*	*	*	
2013	Zhao et al. (2013)					*		*		*		*		*				*		*	Min	*			*		*		*		*		*	*	*	*	
2005	Lee et al. (2005)		*			*		*		*		*		*				*		*	Min			*			*		*		*		*	*	*	*	
2015	Bian et al. (2015)		*			*		*		*		*		*				*		*	Min				*		*		*		*		*	*	*	*	
2015	Ji et al. (2015)		*			*		*		*		*		*				*		*	Min				*		*		*		*		*	*	*	*	
2010	Mäkelä and Wächter (2010)		*			*		*		*		*		*				*		*	Min	*			*		*		*		*		*	*	*	*	
2012	Legido and Mazza (2013)		*			*		*		*		*		*				*		*	Min	*		*		*		*		*		*	*	*	*	*	
2001	Stenken et al. (2001)		*			*		*		*		*		*				*		*	Min			*		*		*		*		*		*	*	*	
2004	Kim et al. (2004)		*			*		*		*		*		*				*		*	Min	*		*		*		*		*		*		*	*	*	
2006	Alvarez (2006)		*			*		*		*		*		*				*		*	Min	*		*		*		*		*		*		*	*	*	
2006	Jung and Kim (2006)		*			*		*		*		*		*				*		*	Min	*		*		*		*		*		*		*	*	*	
2008	Alvarez (2008)		*			*		*		*		*		*				*		*	Min	*		*		*		*		*		*		*	*	*	
2012	Hu et al. (2012)		*			*		*		*		*		*				*		*	Min	*		*		*		*		*		*		*	*	*	

Problem Structure = {A: Solution Outcome: (1: Detailed Stowage Plan, 2: Class-based Stowage Plan, 3: Load sequencing plan, 4: Yard Equipment Planning), B: Terminal Aspect: (1: QC Operations, 2: Yard Operations), C: Number of QCs: (1: One, 2: Multiple), D: Equilibrium and Stability constraints: (1: Explicitly, 2: Implicitly, 3: Is not considered), E: Re-handles at yard or bay: (1: Approximation, 2: Exact number, 3: Is not considered), F: Type of yard equipment (1: Straddle carriers, 2: RTGs, RMGSs or Yard cranes, 3: Reach stackers), G: Loading Policy: (1: Sea-to-land, 2: Other policy, 3: Optimal policy), H: Formulation type: (1: Integrated, 2: Hierarchical)}

Objective Function =  $\{T_w$ : Total weighted loading time (handling, waiting, transshipment etc.) or makespan,  $T$ : Tardiness of operations,  $T_y - Q_y$ : Yard-to-quay transport time,  $S_o$ : Number of container re-handles/reshuffles on/for vessel,  $E_q$ : Violation of equilibrium constraints,  $Q_R$ : Quay cranes related objective}

Solution Approaches = {X: Exact Solutions: (1: Novel mathematical models), Y: Decomposition and Exact Algorithms, Z: Heuristic, Simulation, AI approaches: (1: Beam search, 2: Ant colony optimization, 3: Tabu search, 4: Greedy/GRASP heuristics, 5: Local search, 6: Genetic Algorithms, 7: Simulation)}

## 5.4 Conclusions and Future Research Directions

This paper has presented a description of the SLP as a combination of operational stowage planning, load sequencing and equipment management. The relevant literature has been surveyed and a comparison table has been provided. The study shows that, aside from yard equipment scheduling, little work has been done on the optimization of loading operations.

It is worth noticing that the lack of literature, does not only apply to the integrated SLP but also to the planning problems composing it. Many works have appeared in the past two decades on stowage planning, yet very few focus on the interface with the terminal, and those that do, often look at the problem only from the terminal side.

An explanation for the scarce amount of research in this field, could be explained by the general lack of benchmark data. Each paper presents indications of the nature of the data, but no detailed information. It would be beneficial to have a public benchmark for the SLP, which could then also be used for its subproblems.

Of notice, it is also the use of load policies for the container sequencing. Since most of the surveyed papers have industrial collaborations, this tendency could be explained as a lack of interest from the terminals.

With respect to solution techniques, the literature focuses mainly on heuristic methods. However, exact decomposition algorithms might efficiently solve some reasonable sized problems to optimality.

With focus on the loading operations, we see the integration of yard equipment into the planning models as the nearest research challenge. In none of the papers we surveyed (with exception of Vis et al. (2005) which focuses on routing), yard equipment such as straddle carriers are considered as a limited resource. Optimized resource utilization can have a large impact, especially on short-sea terminals, since the now available resources can be utilized to service e.g. hinterland operation.



## CHAPTER 6

# Formulations for ship loading problem with transfer vehicle assignment and scheduling

---

This chapter introduces containership loading problem in seaport container terminals. The management of loading operations, planning of the equipment to use and their scheduling is what we define as the Ship Loading Problem (SLP). We formulate mathematical models and a number of valid inequalities to enhance these formulations. We also propose a method to compute new lower and upper bounds for the problem. Results show that enhancements on the formulations improve the performance significantly and the lower bound procedure obtains many strong bounds in very short computational times. We also test the model for different objective functions. <sup>1</sup>

---

<sup>1</sup>C. Iris, D. Pacino, S. Ropke, "Formulations for ship loading problem with transfer vehicle assignment and scheduling", Technical report, 2016

## **6.1 Introduction**

Maritime freight transport is an important part of the logistics systems. Benefiting from rapid globalization, the containerized freight transport has been growing steadily in the past several decades. Container terminal throughput, totaling worldwide 684.4 million TEUs in 2014 (UNCTAD (2015)), is estimated to increase by 5.1% in 2015, therefore the increasing container handling volumes makes operations planning a more complex and important for container terminals.

Liner shipping companies have adapted to the growth in the transport volumes by increasing the capacity of their fleet, deploying vessels of over 19,000 TEUs. Capacity is, however, not enough. A reliable service requires the goods to arrive on time, and it is here that container terminals play a major role. The increase in vessels size is not acting alone to intensify the pressure on the container terminals. The slow-steaming has increased the time spent in the open sea (Psaraftis and Kontovas (2010)). An outcome of the slow-steaming is that liner shipping companies deploy many more and larger vessels to the strings and this increases the peak of workloads on the container terminals. In order to meet the tight transit times along the strings, they also expect container terminals to minimize the vessel turnaround (handling) times.

Vessel turnaround times might be reduced by deploying more QCs or transport equipment on each vessel, however there is no guarantee of an improvement in the service quality because there is a limit to the number of equipment that can be deployed to a vessel and inefficient management of these new equipment can bring more congestion and deterioration in the overall performance. Considering that QCs and transport equipment are limited resources with high operating costs, terminals should rather optimize the use of these resources.

We refer readers to the literature reviews on decision problems in seaside operations (Bierwirth and Meisel (2010), Carlo et al. (2013), Bierwirth and Meisel (2014), Iris et al. (2015b)), transport operations (Carlo et al. (2014a)) and yard operations (Li and Vairaktarakis (2004), Carlo et al. (2014b)) in terminals. Literature reviews present that there is an ongoing research avenue that merely focuses on decision problems on which terminals have complete control. In parallel, there is a need for flexibility in operations, and possible collaboration with the liner shipping company can bring some flexibility in ship loading related operations.

The efficient loading of containers to the vessel became a more complicated problem due to the increase in vessel size, vessel numbers and complicated technicalities. The high degree of industrial requirements (e.g. lashing patterns,

vessel stress forces and staff working hour regulations) along with all other mentioned challenges, makes the efficient ship loading even more complicated problem. It also often happens that some of containers are ready to be loaded earlier but have to wait since they would be out of the planned load sequence while, some other containers can easily be spread over large areas due to transshipments, missing space, different services or due to delays from feeder ships, tracks and trains. Due to the large number of interactions between the handling equipment, the cargo arrangement on the vessel/yard, and the scarce number of vehicles, most attempts at improving the loading operations should be based on optimization methods.

Some liner shipping companies are aware of this problem and have actively started to adapt their *stowage plans* to be more terminal friendly. A stowage plan describes the arrangement of containers on the vessel. The liner, makes sure the plan results in as few handling costs as possible and that the vessel is sea-worthy once loaded. In recent years there has been a shift on the stowage planning policy which is based on an increasing collaboration between the container terminal and the liner shipper. The liner provides the container terminal with the stowage plan based on container classes (*class-based stowage plan*), but leaves the terminal the freedom of modifying the arrangement of specific containers of the same class (a container class is defined by its port of discharge, physical dimensions, weight, etc.). In this study, we focus on this problem with further considerations of loading equipment management aiming at reducing working hours handling equipment, increasing the QC intensity and meeting the deadlines of the liner shipping company.

The potential time and cost savings of this collaboration have been initially studied by Monaco et al. (2014) in which authors did not consider the limited number of loading equipment (transfer vehicle). In this study, we include the assignment and scheduling of transfer vehicles into this stowage planning problem. The contribution of the study is mainly two-fold. First we introduce a new integrated container terminal problem to improve the efficiency of loading operations. We formulate a mathematical model to solve the problem and a number of valid inequalities to improve the formulation. Then we suggest a method to obtain the new bounds for the mentioned problem. We also test the formulation with the different objectives of the terminals. Finally, we report the cost savings obtained by integrating these subproblems. Computational results show that the enhancements on the model significantly improve the performance of the formulation. Although the deeper integration makes the problem harder to solve, there are significant cost savings for the terminal operators. In the liner point of view, the problem definition strongly supports the terminal to stick to the deadline of vessel departures and this ensures more reliable services for the liner.



The remainder of the chapter is organized as follows. Section 6.2 presents relevant literature briefly. Section 6.3 addresses the problem definition. Section 6.4 provides the mathematical model and improvements, while Section 6.5 presents a new method to obtain lower and upper bounds. The results are discussed in Section 6.6 and finally the paper is finalized with conclusion and future research perspectives in the last section.

## 6.2 Relevant literature

The problem studied in this chapter is introduced methodologically in Iris and Pacino (2015). The authors also review each subcomponent of the SLP and present the integration efforts between these subcomponents. Monaco et al. (2014) is one of the first studies that distinguishes between the stowage planning problem (solved by the liner shipper) and the operational refinement of it done by the terminal in the loading operations. Fulfilling requirements of class-based stowage plan and making the final match between containers and container classes is referred to *operational stowage planning problem* which is solved by Monaco et al. (2014). Authors solve the problem through a two-phase tabu search method.

The work presented in Imai et al. (2002) is one of the first stowage planning model that considers the yard operations. In that paper, no distinction is made between different container types or classes. The model formulates the stability of the vessel by a measure and solve the problem with an estimate on the number of yard re-handles. In a later paper Imai et al. (2006), the authors include trim and heeling (longitudinal and transversal listing of the vessel) to the before-mentioned model. The stability constraints are still handled in the objective rather than a constraint. The authors propose a solution approach based on Genetic Algorithms (GAs).

In Ambrosino and Sciomachen (2003), a two-stage stage planning problem is solved. In the first-stage a stowage planning model is solved and the generated stowage plan is evaluated in terms of yard re-handles using two policies namely the pre-marshalling and the sort-and-store. The model distinguishes between container type, weight, discharge port. Vessel stability is heuristically handled by balancing the front-back and right-left side of the vessel (an in-depth description of the model can be found in Ambrosino et al. (2004)).

Steenken et al. (2001) focus on the generation of a stowage plans with respect to class-based plan and they dispatch/schedule straddle carriers (SC) simultaneously for the first time. A just-in-time scheduling model is solved when a group

of container is ready to be retrieved from the yard. The model assigns each container to a specific position and a SC with the objective of the minimizing the lateness of the QCs and the transportation time between the yard and the quay area. To solve the problem, authors present a mathematical model and a best-fit heuristic.

Kim et al. (2004) focus on the load sequencing and TC scheduling with respect to operational stowage planning. The presented non-linear mathematical model incorporates vessel stability considerations by imposing weight and height limit constraints on the stacks of the vessel. The sequencing is part of the model decisions, however, a column-wise policy is encouraged with the objective. Only a high travel time of the TCs can force the sequence away from this policy. The actual schedule for the loading of the containers is also modelled with objective of minimizing reshuffles, TCs travel times and interferences. A two stage approach is used to solve the problem. The first stage sequences yard-clusters (like of Lee et al. (2005)), while the second sequences individual containers. Both stages are solved using beam search.

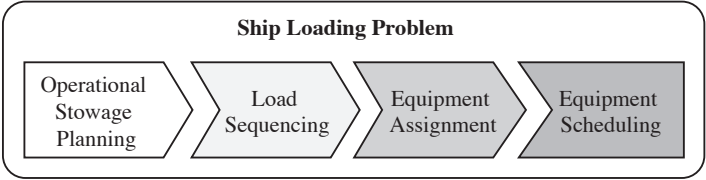
Alvarez (2006) suggests an integration approach that combines detailed reach-stackers planning and scheduling with operative stowage planning. It is assumed that either column-wise or layer-by-layer load sequencing is selected. The model minimizes the number of re-handling, the movement of reach-stacker (i.e. distance traveled on the ground by the transport vehicles) and vessel instability. A solution of the model establishes which sections of the yard provide the containers of the needed type (with given numbers of containers) and generates a feasible tour of the yard to pick them up. A Tabu Search algorithm is described for solving the problem. A Lagrangian relaxation approach is proposed by the same author in a later work Alvarez (2008).

Regardless of operative stowage planning problem, Jung and Kim (2006) have focused on the integration of load scheduling and equipment assignment problems. The authors focus on a YC based container terminal with the objective of minimizing makespan and they solve the problem with a GA. Grunow et al. formulate a dispatching problem for terminals using Automated Guided Vehicles (AGVs) and the performance of the dispatching strategies is evaluated using a scalable simulation model.

## 6.3 Ship Loading Problem

As Figure 6.1 illustrates, we define the SLP as the integration of four terminal planning problems: operational stowage planning, load sequencing, equipment

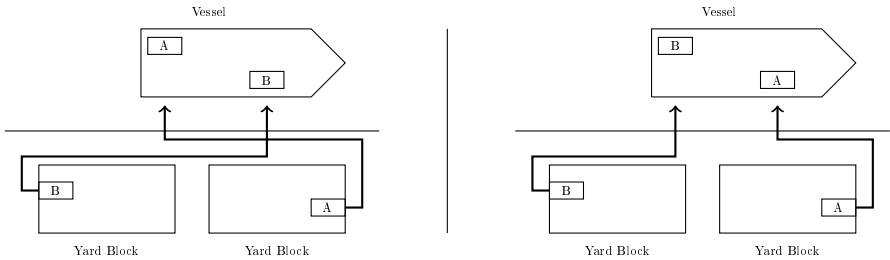
assignment and equipment scheduling. This problem is configured for a single vessel, but it could be extended for multiple ships.



**Figure 6.1:** Ship Loading Problem composition (Iris and Pacino (2015))

We now detail the SLP by describing each of those problems and their interactions.

In particular, we wish to utilize the flexibility that exists when moving from a class-based stowage plan to an operative stowage plan. Consider the example in Figure 6.2. The figure to the left shows a stowage plan composed of two containers of the same class (say two 40-foot containers weighting between 20 and 22 tons having the same destination). The position in the terminal area is also shown and the arrows represent the travel distance needed to bring the container to the vessel. In the figure on the right side, the terminal has re-arranged the assignment of containers with the same container class thus achieving a better plan that requires less travel time. Such flexible assignment helps to integrate the remaining problems into the SLP. The new plan is of no consequence for the liner so long as the container classes are not changed. Summing up, the operative stowage plan deals with assigning specific containers in the yard for each position on the vessel with respect to the class-based stowage plan.



**Figure 6.2:** Class based stowage plan example.

The second component of the SLP is the load sequencing problem. The sequence in which containers will be loaded is firstly governed by physical rules, such as the sequence between two containers destined to the same row cannot be changed due to given QC work-schedule and physical limitation of the vessel (e.g. below

tiers in the same bay/row combination should be loaded before, hatch cover, etc.). It is, however, up to the terminal to decide the sequence of containers on different rows or tiers. The ordering is affected by possible ready time of each container in front of the respective QC and this depends on various factors such as container locations in yard, the availability of handling equipment, the operative stowage plan, etc. The terminal might reduce the total loading time by efficiently ordering containers to load on the vessel.

The load sequencing problem is constrained by QC work-schedule. The QC work-schedule is the set of decisions and it includes the QC assignments (QC split) to bays of the vessel and the loading order between the bays. The QC work-schedule is mostly determined in earlier stages with berth allocation and QC assignment (See Iris et al. (2015b)). In this study, in order not to increase the computational complexity, we assume that QC work-schedule is ready and there is a fixed loading policy for each QC. It is assumed that loading "from-sea-to-land" with "stack-wise" sequencing is applied for each QC.

The SLP finally covers the assignment and scheduling of loading equipment (i.e. transfer vehicles). Integrating the transfer vehicles (TVs) into the SLP is vital because they are limited resources in the terminal and they might cause a bottleneck in loading operations. What is more, generating a feasible schedule of transfer vehicles will determine the ready time of each container in front of the respective QC more accurately. These ready times influence the assignments of containers to each position. The SLP studied in this chapter covers the assignment of specific TVs to each QC and the schedule of all TVs to load all containers to the vessel. In the other words, the problem deals with determining which specific container will be picked up by which TV at what point in the time. It is assumed that the number of TVs that works on each QC might change over time (time-variant assignment). This means that for example, a solution can hold 3 TVs working on a QC for an hour then it can be reduced to 2 TVs for the remaining loading time.

The efficient planning of the SLP might reduce port stay times, the number of transport vehicle required to load and unload the vessel, terminal related emissions, the idle time of QCs, etc.

The class-based stowage plan and the QC work-schedule have been decided and are inputs for the SLP in this study. The problem definition is based on further assumptions which will detailed as follows:

- It is assumed that unloading operations are performed first, then loading operations start (i.e. the problem does not include dual cycling operations).

- It is assumed that a loading policy for each QC is determined beforehand. This means that the sequence, in which each position is loaded, is known. It is still not known which specific container will be loaded to each position.
- It is assumed that the retrieval order of containers in the yard does not generate any yard-shifts within the yard bay (e.g. a variant of pre-marshalling problem is solved beforehand)
- The stability of vessel is ensured with the class-based stowage-plan.
- Each TV can only work for a single QC during the loading of the vessel. In the other words, it is not allowed to pool TVs for each QC.
- It is assumed that TV operations are non-preemptive. When a TV is assigned to a QC it does not stop until it finishes its tasks on that QC.
- It is assumed that the congestion in the yard, the travel speed of a TV with/without a container are all reflected in the transportation times between yard positions (or I/O point depending on the available yard type) and QCs.
- It is assumed that there is no buffer for any containers under the QC, then the TV and QC operations are not decoupled.

The SLP in this study aims at determining the assignment of each container to a vessel position (slot). It also determines which TV will load each container to their positions, and the time of pickup from the yard block and dropping in front of QC is also decided (i.e. the complete schedule of all TVs are also made).

## 6.4 Mathematical Model for the SLP

The list of notations, i.e. parameters, decision variables is as follows:

Table 6.1: SLP mathematical notation

Parameters and sets:	
$C$	Set of containers that will be loaded to vessel
$C_p$	Set of containers belonging to a class suitable for slot position $p$
$Q$	Set of quay cranes that are assigned to load the vessel
$Q_p$	Set of quay cranes that loads the container for position $p$ , one element set
$P$	Set of positions to be served
$P_i$	Set of positions that matches with the class type of container $i$
$P_q$	Set of positions that will be loaded by QC $q$
$P_p^{crane}$	Set of positions that are handled by the same crane as position $p$
$S$	Set of transfer vehicles available to serve the vessel
$S_p$	Set of transfer vehicles that can serve position $p$
$S_q$	Set of transfer vehicles that are available to serve QC $q$
$T$	Set of time periods, $T \in \{0, 1, \dots, H - 1\}$ , where $H$ is the closing of planning horizon
$\tau_{ip}$	The time needed for a transfer vehicles to transport container $i$ from its yard-position to vessel-position $p$ . The time needed is assumed to be equal in both directions.
$\beta$	The handling time for each QC, minimum time between two consecutive container loading operation
$EFT$	Expected finishing time of operations for the vessel
$\alpha$	The cost of using one TV for one time-unit
$\gamma$	The cost of exceeding the expected finishing time ( $EFT$ ) for one time-unit
$M$	A large positive number
Decision variables:	
$t_p^s \in \mathbb{Z}^+$	Time when container for position $p$ has been dropped in front of QC by TV $s$ (container dropping time)
$Start_s \in \mathbb{Z}^+$	Time when operations of TV $s$ starts
$End_s \in \mathbb{Z}^+$	Time when operations of TV $s$ ends
$z \in \mathbb{Z}^+$	Makespan for the operations
$\Delta EFT \in \mathbb{Z}^+$	Lateness of operations
$x_{ip}^s \in \mathbb{B}$	1; if the container $i$ is loaded to position $p$ picked up by TV $s$ , 0 otherwise

Let us now introduce the mathematical model:

$$\min \alpha \sum_{s \in S} (End_s - Start_s) + \gamma \Delta EFT \quad (6.1)$$

subject to

$$\sum_{p \in P_i} \sum_{s \in S_p} x_{ip}^s = 1 \quad \forall i \in C \quad (6.2)$$

$$\sum_{i \in C_p} \sum_{s \in S_p} x_{ip}^s = 1 \quad \forall p \in P \quad (6.3)$$

$$2\tau_{ip} - M(2 - x_{ip}^s - \sum_{i \in C_{p'}} x_{ip'}^s) \leq t_p^s - t_{p'}^s \quad \forall i \in C, \forall s \in S, \forall p \in P_i, p' \in P_p^{crane} \mid p' \prec p \quad (6.4)$$

$$\sum_{s \in S} t_p^s \geq \sum_{s \in S} t_{p'}^s + \beta \quad \forall p \in P, p' \in P_p^{crane} \mid p' \prec \prec p \quad (6.5)$$

$$t_p^s \leq \sum_{i \in C_p} H x_{ip}^s \quad \forall p \in P, \forall s \in S_p \quad (6.6)$$

$$t_p^s \geq 2 \sum_{i \in C} \tau_{ip} x_{ip}^s \quad \forall p \in P, \forall s \in S_p \quad (6.7)$$

$$t_p^s - 2 \sum_{i \in C} \tau_{ip} x_{ip}^s + H(1 - \sum_{i \in C} x_{ip}^s) \geq Start_s \quad \forall s \in S, \forall p \in P \quad (6.8)$$

$$t_p^s \leq End_s \quad \forall s \in S, \forall p \in P \quad (6.9)$$

$$Start_s \leq End_s \quad \forall s \in S \quad (6.10)$$

$$z \geq t_p^s + \beta \quad \forall s \in S, \forall p \in P \quad (6.11)$$

$$x_{ip}^s = 0 \quad \forall i \in C, \forall p \in P, \forall s \in S \setminus S_p \quad (6.12)$$

$$\Delta EFT \geq z - EFT \quad (6.13)$$

$$t_p^s, Start_s, End_s, z, \Delta EFT \in \{0, \dots, H - 1\} \quad \forall s \in S, \forall p \in P \quad (6.14)$$

$$x_{ip}^s \in \{0, 1\} \quad \forall i \in C, \forall s \in S, \forall p \in P \quad (6.15)$$

The model has decision variables regarding operative stowage plan and TV assignment to containers ( $x_{ip}^s$ ), TV scheduling ( $t_p^s, Start_s, End_s$ ). There are also other auxiliary variables which help to obtain link between operative stowage plan and TV management.

The objective function is a combination of the cost of TV-times and the lateness (if loading finishes after expected finishing time). Constraint (6.2) ensures that each container will be loaded to a position that matches with its container class. Constraint (6.3) guarantees that all positions will be loaded with a container that matches the container class of that position. For a given container, TV and position, constraint (6.4) makes sure that the container dropping time for that position is set correctly. This is done by forcing the difference between two consecutive positions' dropping times to be equal or greater than the time required to bring the container ( $2\tau_{ip}$ ) to the QC. The term multiplied by  $M$  on left-hand side makes sure that the constraint is only active when the two positions which are following each other in loading order are handled by the same TV. Constraint (6.5) ensures that all positions are loaded in the correct order, and the containers that will arrive at the same QC should have at least  $\beta$  time apart. Note that  $p' \prec\prec p$  is meant that the position  $p'$  is handled immediately before position  $p$  according to the loading policy, while  $p' \prec p$  shows that position  $p'$  is loaded before position  $p$ . Constraint (6.6) ensures that if there is a TV assignment to a position, then the container dropping time is earlier than end of planning horizon. Assuming that each TV is in front of its respective QC in the initial position, constraint (6.7) guarantees that that earliest dropping time is transportation time of the container which is picked up. Constraint (6.8) sets the starting time of each TV, while constraint (6.9) sets the ending time of each TV operations. If a TV is not assigned to a QC, these variables take a value of zero. Constraint (6.10) is the link between start and end time for each TV operation. Constraint (6.11) obtains the makespan, while constraint (6.12) ensures that a container for position  $p$  cannot be picked up by TV  $s$ , if it is not assigned to serve this position. Constraints (6.14)-(6.15) determine the domain of variables.

### 6.4.1 Enhancements for the SLP model

This section introduces a number of enhancements which are based on formulating lower bounds on variables and valid inequalities for the SLP.

#### 6.4.1.1 Lower bounds for variables

Before detailing the inequalities, let us formulate the minimum transportation time that is required in total to transfer containers that will be loaded by QC  $q$ . That is obtained when the nearest container is picked up for all positions that will be loaded by that QC. We go through each position, which is in the loading order of QC  $q$ , one by one and pick up the container with the least



transportation time ( $2\tau_{ip}$ ) for that position. If one specific container is picked up for an earlier position of the same QC, we do not consider it again. The minimum transportation time for each of the selected container is called with parameter  $\tau_{ip}^{min}$ . The minimum overall time required to transport all containers of a specific QC  $q$  ( $\delta_{min}^q$ ) is then:

$$\delta_{min}^q = \sum_{i \in C_p} \sum_{p \in P_q} 2\tau_{ip}^{min} \quad \forall q \in Q \quad (6.16)$$

Now suppose that we have a feasible schedule for all TVs. Then there exists a feasible schedule with the same makespan in which all waiting times of each TV occur just before it drops the container in front of the QC. We now formulate a lower bound on the total waiting time in front of QC  $q$ . To formulate it, we first need the maximum transportation time to load a container to a given position matching with its container class. We have to compare this value with the QC loading time  $\beta$ . If the QC loading time is longer than maximum interarrival time for the container, this means TVs must wait under the QC  $q$  for the difference between  $\beta$  and interarrival time. When we sum up the waiting time for all positions that will be loaded by QC  $q$ , we obtain a lower bound on the total TV waiting time for the QC  $q$  as (6.17).

$$LB_{waiting}^q = \sum_{p \in P_q} \max \left\{ 0, (\beta - 2 \max_{c \in C_p} \{\tau_{cp}\}) \right\} \quad \forall q \in Q \quad (6.17)$$

We now set lower bounds on various variables. First an inequality is formulated, and it ensures that the makespan should be larger than the maximum of finishing times of all QCs. The lower bound on the makespan of each QC is obtained as the maximum of the total loading time ( $\beta|P_q|$ ) and the minimum TV transportation/waiting time for QC  $q$  where  $\left\lceil \frac{\delta_{min}^q}{|S_q|} \right\rceil$  is the lower bound on the minimum transportation time for each TV to load all containers of QC  $q$ . We formulate Constraint (6.18) for this lower bound.

$$z \geq \max_{q \in Q} \left\{ \max\{\beta|P_q|, \left\lceil \frac{\delta_{min}^q}{|S_q|} \right\rceil + LB_{waiting}^q\} \right\} \quad (6.18)$$

Since we know that minimum  $\delta_{min}^q$  TV-time is required to transport containers to all positions that will be loaded by QC  $q$  and the minimum waiting time for

QC  $q$  is  $LB_{waiting}^q$ , we formulate a lower bound on the TV working times that are required to serve QC  $q$  as:

$$\sum_{s \in S_q} (End_s - Start_s) \geq \delta_{min}^q + LB_{waiting}^q \quad \forall q \in Q \quad (6.19)$$

Another lower bound is set on the dropping time ( $t_p^s$ ) variables. For each position  $p$ , the  $t_p^s$  is at least the maximum of the total loading time of all positions before  $p$  and the minimum transportation time required to load all positions before  $p$ . The minimum transportation time is obtained in a similar way to  $\tau_{ip}^{min}$ . A set named  $B_p$ , which holds containers of given class that are loaded before position  $p$ , is generated on fly. If a container is picked up before, it is not reevaluated to be placed to another position. Since all positions in the precedence relationship are loaded by the same QC, they share the same pool of TVs. Constraint (6.20) sets the lower bound.

$$\sum_{s \in S_p} t_p^s \geq \max \left\{ \sum_{p' \in p' \prec p} \left\lceil \frac{\min_{i \in C_{p'}, i \notin B_{p'}} \{2\tau_{ip'}\}}{|S_{p'}|} \right\rceil, \sum_{p' \in p' \prec p} \beta \right\} \quad \forall p \in P \quad (6.20)$$

#### 6.4.1.2 Valid inequalities for SLP model

The first set of valid inequalities focuses on the container classes rather than specific containers. Let us call the set of all container classes as  $U$  and the set of container classes for each QC  $q$  as  $U_q$ . We can extract the set of containers which has a class of  $u$  ( $C_u$ ) and the set of positions which requires a container class  $u$  that will be loaded by QC  $q$  ( $P_{qu}$ ). The valid inequality (6.21) ensures that for the given QC and container class, the class-based stowage plan determines the total number of containers to be assigned to each QC.

$$\sum_{s \in S_q} \sum_{i \in C_u} \sum_{p \in P_{qu}} x_{ip}^s = |P_{qu}| \quad \forall q \in Q, \forall u \in U_q \quad (6.21)$$

The second family of inequalities tries to better link the integer start/end and binary assignment variables. We now introduce a new binary variable  $y_t^s$  which takes value of 1, if TV  $s$  is operating in period  $t$ , 0 otherwise.

$$\sum_{t \in T} y_t^s = End_s - Start_s \quad \forall s \in S \quad (6.22)$$

$$ty_t^s + H(1 - y_t^s) \geq Start_s \quad \forall t \in T, \forall s \in S \quad (6.23)$$

$$(t + 1)y_t^s \leq End_s \quad \forall t \in T, \forall s \in S \quad (6.24)$$

$$\sum_{s \in S_q} y_t^s \leq |S_q| \quad \forall t \in T, \forall q \in Q \quad (6.25)$$

$$\sum_{t \in T} \sum_{s \in S_q} y_t^s \geq \delta_{min}^q \quad \forall q \in Q \quad (6.26)$$

$$y_t^s \in \{0, 1\} \quad \forall s \in S, \forall t \in T \quad (6.27)$$

Constraint (6.22) ensures that for each TV, the sum of  $y_t^s$  for all periods should be equal to the operating time of that TV. Constraints (6.23)-(6.24) guarantee that, the  $y_t^s$  takes a value of one in periods between starting and ending times. Constraint (6.25) sets the maximum number of TVs that can be assigned to each QC for each period, while constraint (6.26) guarantees that at least  $\delta_{min}^q$  TV-time is needed to transport all containers to QC  $q$ . Finally the domain of  $y_t^s$  is set in constraint (6.27). In order to reduce the number of variables and constraints, we formulate an upper bound on  $H$ . Simply by assuming that only one QC and one TV are used, we can obtain an upper bound on the planning horizon by obtaining maximum processing and waiting time,  $H = \max \left\{ \sum_{p \in P} \left\{ 2 \max_{c \in C_p} \{\tau_{cp}\} + \max \{0, (\beta - 2 \min_{c \in C_p} \{\tau_{cp}\})\} \right\}, \beta |P| \right\}$ .

Finally we formulate a valid inequality that better links the assignment and scheduling variables. Constraint (6.28) ensures that the total time that a TV will be operating is at least the total transportation time to load containers that are going to be loaded by that TV.

$$End_s - Start_s \geq 2 \sum_{i \in C_p} \sum_{p \in P_i} \tau_{ip} x_{ip}^s \quad \forall s \in S \quad (6.28)$$

We call the enhanced version of the SLP model as SLP+.

## 6.5 New lower and upper bounds for the SLP

In order to obtain updated lower bounds for the SLP rapidly, we focus on the components of the objective function. It composes of two parts: the cost of total TV-time and the cost of ending later than expected finishing time.

We formulate a new mathematical model that omits decision variables related to TV scheduling ( $t_p^s, Start_s, End_s$ ) and this model obtains a lower bound on the SLP. Let us first show that we can obtain lower bounds on objective components by solely using  $x_{ip}^s$  variables.

**Proposition 1:**  $\sum_{i \in C} \sum_{p \in P_i} \sum_{s \in S_p} 2\tau_{ip}x_{ip}^s$  is a lower bound on  $\sum_{s \in S} (End_s - Start_s)$ .

Proof: The proof is evident when we note that  $\sum_{i \in C} \sum_{p \in P_i} \sum_{s \in S_p} 2\tau_{ip}x_{ip}^s$  is the exact transportation time of all TVs, while  $\sum_{s \in S} (End_s - Start_s)$  holds the exact transportation time and the TV waiting times which are proved to be non-negative by the definition of lower bound  $LB_{waiting}^q$ . Note that  $End_s, Start_s$  variables take a value of zero for unused TVs.  $\square$

**Proposition 2:**  $\beta + \max_{s \in S} \left\{ \sum_{i \in C} \sum_{p \in P_i} 2\tau_{ip}x_{ip}^s \right\}$  is a lower bound on  $z$ .

Proof: The definition of makespan for the SLP states that makespan is bounded by the maximum of the finishing times of all TVs added by the loading time of one last container. Then we have to show that  $\sum_{i \in C} \sum_{p \in P_i} 2\tau_{ip}x_{ip}^s$  is a lower bound on the maximum finishing time of TV  $s$ . The finishing time of TV  $s$  is at least the summation of all transport times for containers that it will load to its QCs.  $\square$

**Proposition 3:**  $\max_{q \in Q} \left\{ \beta |P_q| \right\}$  is a lower bound on  $z$ .

Proof: The definition of makespan for the SLP states that makespan is bounded by the maximum of the finishing times of all QCs that work on the vessel. The finishing time of one QC is at least loading time of all positions that the QC is assigned to ( $\beta |P_q|$ ).  $\square$

The model that we suggest to obtain the lower bound on the SLP uses the same notation and variables of the SLP model. We introduce a new integer variables;  $TTS_s$  which presents the lower bound on the finishing time of operations for TV  $s$ . Now, let us introduce the new model which is called lower bounding model refereed to as LB-SLP:

$$\min \alpha \sum_{i \in C} \sum_{p \in P_i} \sum_{s \in S_p} 2\tau_{ip} x_{ip}^s + \gamma \Delta EFT \quad (6.29)$$

subject to

$$\sum_{p \in P_i} \sum_{s \in S_p} x_{ip}^s = 1 \quad \forall i \in C \quad (6.30)$$

$$\sum_{i \in C_p} \sum_{s \in S_p} x_{ip}^s = 1 \quad \forall p \in P \quad (6.31)$$

$$x_{ip}^s = 0 \quad \forall i \in C, \forall p \in P, \forall s \in S \setminus S_p \quad (6.32)$$

$$TTS_s = \beta + \sum_{i \in C} \sum_{p \in P_i} 2\tau_{ip} x_{ip}^s \quad \forall s \in S \quad (6.33)$$

$$z \geq TTS_s \quad \forall s \in S \quad (6.34)$$

$$z \geq \beta |P_q| \quad \forall q \in Q \quad (6.35)$$

$$z \geq \left\lceil \frac{\delta_{min}^q}{|S_q|} \right\rceil + LB_{waiting}^q \quad \forall q \in Q \quad (6.36)$$

$$\Delta EFT \geq z - EFT \quad (6.37)$$

$$TTS_s, TTQ_q, z, \Delta EFT \in \{0, \dots, H - 1\} \quad \forall s \in S \quad (6.38)$$

$$x_{ip}^s \in \{0, 1\} \quad \forall i \in C, \forall s \in S, \forall p \in P \quad (6.39)$$

Any feasible solution to (6.29)-(6.39) will be a lower bound on the SLP. The objective function (6.29) is a combination of the cost of TV-times and the cost of being late. Constraints (6.30)-(6.32) are interpreted in a similar way to SLP model. Constraint (6.33) sets the lower bound on the finishing time for each TV, constraint (6.34) uses these variables to obtain the makespan for the vessel. Constraint (6.35) sets the lower bound on finishing time for each QC, where  $|P_q|$  refers to the number of positions that will be loaded by QC  $q$ . Constraint (6.36) uses minimum transportation time to obtain the lower bound on the makespan for the vessel. Constraints (6.37)-(6.39) are similar to the SLP model.

The properties of the SLP allows us to obtain upper bounds with the solution obtained from the lower bounding model (LB-SLP). The assignment decisions

made in the LB-SLP will always generate feasible solutions for the SLP. This is because the remaining problem is a TV scheduling problem with given assignments and transportation times. We use this remainder problem to obtain upper bound for the SLP. Let us call the assignment decisions which are obtained from the LB-SLP as  $x_{ip}^{\hat{s}}$ . They are now parameters for the following model. The remainder problem inherits the notation and objective function of the SLP. We now introduce the model to obtain the upper bounds and it is referred as UB-SLP.

$$\min \alpha \sum_{s \in S} (End_s - Start_s) + \gamma \Delta EFT \quad (6.40)$$

subject to

$$(6.2) - (6.13) \quad (6.41)$$

$$x_{ip}^s = x_{ip}^{\hat{s}} \quad \forall i \in C, \forall s \in S, \forall p \in P \quad (6.42)$$

$$t_p^s, Start_s, End_s, z, \Delta EFT \in \{0, \dots, H - 1\} \quad \forall s \in S, \forall p \in P \quad (6.43)$$

$$x_{ip}^s \in \{0, 1\} \quad \forall i \in C, \forall s \in S, \forall p \in P \quad (6.44)$$

The objective function and set of constraints can be read in the same way of the SLP model. Constraint (6.42) sets  $x_{ip}^s$  variables to  $x_{ip}^{\hat{s}}$  values.

## 6.6 Computational analysis

We now analyze the performance of each formulation, valid inequalities and new bounds on a set of benchmark instances. All models are run on a 32 core AMD Opteron at 2.8Ghz and 132Gb of RAM computer, and computational times are reported in seconds.

The benchmark includes six sets of test instances. Each instance has either 60 or 240 containers, while the number of QCs differs between 2, 4 and 6. The number of TVs is proportional to the number of QCs, we assume that 3 TVs are available to each QC. The number of positions that will be loaded by each QC is also proportional to the number of QCs. Giving an example, assume that 240 containers will be loaded by 4 QC, this means that each QC loads 60 containers with a pool of 3 TVs. The parameters  $\alpha, \gamma$  are selected to be 5 and 30, respectively. We test the methods for two versions of EFT, one setting assumes that a no-tight EFT is selected (300 minutes), while in the other settings, tight EFT (40, 75 minutes) values are selected. The loading time  $\beta$  is assumed to be

2 minutes, while transportation times are generated according to the layout of the yard and container positions. The number of container classes also differs. For 60 containers instances, it is assumed that 25 container classes are given, while there are 48 container classes in the instances with 240 containers.

All models are solved using CPLEX 12.6.1 solver. A time limit of 5 hours is imposed to solve the SLP and SLP+ models with the options of emphasizing optimality and aggressive cuts. In default conditions, models are run with 4 threads. In the case of memory overflows, they are run with a single thread.

### 6.6.1 Results

Before presenting the detailed results for the SLP, SLP+ models and the new bounds, we first analyze the effects of lower bounds on variables and valid inequalities to the SLP model. Table 6.2 and 6.3 present the value of linear programming (LP) relaxation lower bounds and the CPU times to obtain these lower bounds for each instance in the tight EFT set, respectively. Table 6.2 and 6.3 are composed of two main parts, one part with results of the LP relaxation, and the other with results which are gathered after CPLEX cutting planes are added to the root node. In Table 6.2, the first column presents the EFT for the instance. The next columns  $|C|$ ,  $|S|$ ,  $|Q|$  present the number of containers, TVs and QCs, respectively. The column  $LB_r$  points out the lower bound without any enhancement, while the remainder of columns include different enhancements indexed with the equation number. The unique difference of Table 6.3 from Table 6.2 is that Table 6.3 holds CPU times to obtain the mentioned lower bounds of Table 6.2.

Table 6.2 suggests that lower bounds have been improved for almost all instances, while the lower bounds of are improved better than the remaining valid inequalities with the use of (6.18-6.20) and (6.28). Table 6.2 presents that constraints (6.21) and (6.28) improve the lower bound better after the inclusion of CPLEX cutting planes. These comments hold for all enhancements, except (6.23)-(6.27). Although these constraints marginally contribute to the LBs, they increase the number of constraints significantly and worsen some lower bounds. Additionally, these inequalities increase the computational effort without entailing markable gains. For such reasons, constraints (6.23)-(6.27) will not be used in the SLP+ model. Table 6.3 points out that imposing the enhancements except (6.23)-(6.27) does not significantly increase the computational effort. Hence we justify the use of remaining enhancements with the help of Table 6.2 and 6.3.

Table 6.2: The LP relaxation lower bounds

					CPLEX cutting planes disabled				CPLEX cutting planes active				
EFT	C	S	Q	$LB_r$	(6.18-6.20)	(6.21)	(6.22)-(6.27)	(6.28)	$LB_r$	6.18-6.20	(6.21)	(6.22)-(6.27)	(6.28)
40	60	6	2	0.0	2130.0	0.0	0.0	1890.0	0.0	2130.0	0.0	260.0	2670.0
	60	12	4	0.0	1530.0	0.0	0.0	1890.0	1.4	1530.0	0.0	120.0	1890.0
	60	18	6	0.0	1530.0	0.0	0.0	1890.0	0.1	1530.0	0.2	0.1	1890.0
75	240	6	2	250.0	9490.0	250.0	250.0	7110.0	1904.3	9490.0	2400.0	1904.3	8870.0
	240	12	4	0.0	5890.0	0.0	0.0	6860.0	110.0	5890.0	110.0	0.0	6860.0
	240	18	6	0.0	4690.0	0.0	0.0	6860.0	0.0	4690.0	140.0	70.0	6860.0

Table 6.3: The CPU time to obtain lower bounds

					CPLEX cutting planes disabled					CPLEX cutting planes active			
EFT	C	S	Q	$LB_r$	(6.18-6.20)	(6.21)	(6.22)-(6.27)	(6.28)	$LB_r$	6.18-6.20	(6.21)	(6.22)-(6.27)	(6.28)
40	60	6	2	0.3	0.4	0.4	0.5	0.3	4.1	5.1	4.3	4.4	4.1
	60	12	4	0.2	0.3	0.2	0.4	0.4	3.2	3.4	3.1	3.5	2.5
	60	18	6	0.1	0.4	0.3	0.2	0.4	2.1	2.5	2.6	2.9	1.8
75	240	6	2	91.1	92.1	96.3	212.4	106.2	463.2	482.1	469.7	773.5	891.1
	240	12	4	35.9	37.3	33.8	60.4	41.1	258.1	266.2	193.1	250.5	334.0
	240	18	6	32.7	12.7	29.2	21.2	33.6	121.2	69.2	92.0	225.3	156.8



Results for the SLP, SLP+, LB-SLP and UB-SLP models are presented in Table 6.4 and 6.5. In these tables, the columns identifying the instance are similar to Table 6.2. The columns named "UB" show the best upper bound obtained, while "LB" report the best lower bounds found with that method. Results are indexed with different numbers, the index 1 presents the SLP model results, while index 2 refers to the SLP+ model, and finally 3 reports the bounds that are obtained with the LB-SLP and the UB-SLP. The optimality gap (Gap (%)) is calculated between the each UB and LB ( $\text{Gap} = \frac{UB-LB}{UB}$ ). The LB-SLP and UB-SLP are models that do not guarantee an optimal solution, hence gaps of new lower and upper bounds ( $G_u, G_l$ ) by the LB-SLP and UB-SLP are calculated with the best known upper and lower bounds (from SLP, SLP+) and new upper and lower bounds (LB-SLP, UB-SLP) for each instance ( $G_u = \frac{UB_3 - \min\{UB_1, UB_2\}}{UB_3}$ ,  $G_l = \frac{\max\{LB_1, LB_2\} - LB_3}{LB_3}$ ). Finally tables hold information about the CPU time,  $T$  represents the time to obtain each bound in seconds.

The first important observation is that the SLP model does not perform well for any instance size. Speaking of EFT=300, results show that the SLP model obtains an average of 74% and 98% optimality gaps for 60, 240 containers, respectively. These gaps are 66% and 91% for EFT=40,75 instances. Whenever enhancements are imposed to the SLP model, the performance significantly increases. The SLP+ model solves instances to optimality for 60 containers for EFT=300, while it obtains a gap of 17% for 240 containers for EFT=40,75 instances. The average improvement from SLP to SLP+ is 67% for 60-container instances, while it is 73% for 240-container instances. The SLP+ model can solve four instances to optimality within time-limits and it improves all upper and lower bounds compared to the SLP model.

The lower bounds obtained by the LB-SLP are as good as the result of the SLP+ model, while the upper bounds by the UB-SLP have an average gap of 31% and 38% (from best upper bounds of SLP, SLP+ models) for 60, 240-container instances, respectively. These methods aim at obtaining upper and lower bound in a rapid fashion, so the average time to obtain these bounds is less than one second for 60-container instances, while it is around 2 seconds for the 240-container instances.

**Table 6.4:** Computational results for EFT:300

	SLP				SLP+				Bounds with LB-SLP and UB-SLP					
	$ C ,  Q ,  S , \text{EFT}$	$UB_1$	$LB_1$	Gap	$T_1$	$UB_2$	$LB_2$	Gap	$T_2$	$UB_3$	$LB_3$	$G_u$	$G_l$	$T_{UB_3}$
	60,2,6,300	1890	400.0	0.79	18000	1890	1890.0	0.00	4420	2430	1890	0.22	0.00	0.8
	60,4,12,300	1890	501.1	0.73	18000	1890	1890.0	0.00	37	2520	1890	0.25	0.00	0.2
	60,6,18,300	1890	566.5	0.70	18000	1890	1890.0	0.00	6	2210	1890	0.14	0.00	0.6
	240,2,6,300	13550	50.1	1.00	18000	12130	6860.0	0.43	18000	19100	6860	0.36	0.00	0.2
	240,4,12,300	7910	120.1	0.98	18000	7610	6860.0	0.10	18000	12930	6860	0.41	0.00	0.6
	240,6,18,300	7520	178.5	0.98	18000	7310	6860.0	0.06	18000	12720	6860	0.43	0.00	1.0

**Table 6.5:** Computational results for EFT:40,75

	SLP				SLP+				Bounds with LB-SLP and UB-SLP					
	$ C ,  Q ,  S , \text{EFT}$	$UB_1$	$LB_1$	Gap	$T_1$	$UB_2$	$LB_2$	Gap	$T_2$	$UB_3$	$LB_3$	$G_u$	$G_l$	$T_{UB_3}$
	60,2,6,40	3240	1450	0.55	18000	3230	2676.0	0.17	18000	5430	2670	0.41	0.00	0.2
	60,4,12,40	2030	586.1	0.71	18000	1970	1890.0	0.05	18000	4000	1890	0.51	0.00	0.07
	60,6,18,40	1890	491.1	0.74	18000	1890	1890.0	0.00	23	2830	1890	0.33	0.00	0.1
	240,2,6,75	17380	2563.2	0.85	18000	16200	11870.2	0.27	18000	24570	11810	0.29	0.01	0.43
	240,4,12,75	12690	937.7	0.93	18000	10800	8270.1	0.23	18000	17570	8210	0.39	0.01	0.6
	240,6,18,75	9080	415.1	0.95	18000	8760	7160.0	0.18	18000	14570	7010	0.40	0.02	0.31

### 6.6.2 Hierarchical vs integrated planning

We further investigate the cost savings with the integration of the operative stowage planning and TV assignment/scheduling, and compare results to the hierarchical planning method. The hierarchical planning is simulated as follows. We first solve the operative stowage planning problem with the objective function of minimizing the transportation times of all containers (Monaco et al., 2014).

$$\min \sum_{i \in C} \sum_{p \in P_i} \sum_{s \in S_p} 2\tau_{ip} x_{ip}^s \quad (6.45)$$

subject to

$$\sum_{p \in P_i} \sum_{s \in S_p} x_{ip}^s = 1 \quad \forall i \in C \quad (6.46)$$

$$\sum_{i \in C_p} \sum_{s \in S_p} x_{ip}^s = 1 \quad \forall p \in P \quad (6.47)$$

$$x_{ip}^s \in \{0, 1\} \quad \forall i \in C, \forall s \in S, \forall p \in P \quad (6.48)$$

The model and constraints are interpreted as in the SLP model. After the  $x_{ip}^s$  variables are determined with this model, we supply them as parameters to (6.40)-(6.44) model and obtain results of hierarchical planning. We compare the hierarchical planning results with the upper bounds obtained from the SLP+ model and UB-SLP in Table 6.6. We identify the hierarchical planning results with index of 4 in Table 6.6.

Results show that the average actual cost savings ( $\frac{UB_4 - UB_2}{UB_4}$ ) through integration (hierarchical vs SLP+) is 43.3% for EFT=40,75 instances, while it is 26.4% for EFT=300 instances. The actual savings do not have any visible correlation with the instance size. This comparison suggests that there is a big potential for savings for the terminal operators. However, the computational time to obtain the SLP+ model results can be too long. For this reason, we also compare the results of the hierarchical planning with upper bounds obtained with the UB-SLP. For EFT=40,75 instances, upper bounds of the UB-SLP are mostly better than the hierarchical planning. For EFT=300 instances, the hierarchical planning outperforms upper bounds of the UB-SLP. Since no-tight EFT (300) instances give a high flexibility on the makespan, the LB-SLP model cannot dominate the operative stowage planning model of (6.45)-(6.48).

Table 6.6: Integrated vs hierarchical planning of SLP

																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			</
--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	----

### 6.6.3 What is the minimum fleet size?

Some container terminals aim at loading containers with the least amount of TVs. In this problem, the goal is to determine the minimum number of TVs required to transport all the containers before the EFT (See Vis et al. (2005) for a version of this problem for an AGV-operating container terminal). The same assumptions and conditions of the SLP hold for the minimum fleet size problem. The only different condition is that EFT bounds the makespan as a hard constraint rather than a soft constraint.

We formulate a variant of the SLP+ model to solve this problem. We introduce a new binary variable  $r_q^s$  which takes the value of one if TV  $s$  works to load QC  $q$ , zero otherwise. Let us now introduce the modified model which uses the same notation of the SLP model.

$$\min \sum_{q \in Q} \sum_{s \in S_q} r_q^s \quad (6.49)$$

subject to

$$(6.2) - (6.12) \quad (6.50)$$

$$(6.18) - (6.21), (6.28) \quad (6.51)$$

$$z \leq EFT \quad (6.52)$$

$$r_q^s \leq \sum_{p \in P_q} x_{ip}^s \quad \forall i \in C, \forall q \in Q, \forall s \in S_q \quad (6.53)$$

$$(6.14) - (6.15) \quad (6.54)$$

$$r_q^s \in \{0, 1\} \quad \forall q \in Q, \forall s \in S \quad (6.55)$$

The objective function (6.49) is the minimization of the number of TVs used to load all containers. Constraint (6.52), which imposes the bound on  $z$ , replaces constraint (6.13). Constraint (6.53) ensures that if a container is picked up by TV  $s$  in order to load to a position by QC  $q$ ,  $r_q^s$  should take a value of one, otherwise due to minimization of  $r_q^s$ , it will take a value of zero. Finally we set the domain of variables. We run the model in the same settings of the SLP+ model. Since EFT imposes a hard constraint on  $z$ , we test different values of EFT. In Table 6.7, we report these results. The columns can be reviewed with the same approach compared to the remaining tables.

**Table 6.7:** Minimum TV fleet size

C	S	Q		EFT	UB	LB		EFT	UB	LB
60	6	2		300	2	2		150	4	4
60	12	4			4	4		80	8	5
60	18	6			6	6		40	12	12
240	6	2			-	2		480	6	2
240	12	4			8	4		240	12	4
240	18	6			6	6		200	11	6

The first important observation is that instances with 60 containers are mostly solved to optimality. For 240 containers, results are more erratic. The gap between upper and lower bounds illustrates that there might be further improvements to load 240 containers with fewer TVs. Considering an EFT of 300 minutes, instances with 60 containers are all solved to optimality. Each QC requires only one TV to load 60 containers in 300 minutes. In order to load 240 containers in 300 minutes, the model cannot find a feasible solution for 2 QCs case; for 6 QCs case, we only need 6 TVs to load 240 containers. Results show that loading 240 containers in 240 minutes by using 4 QC is possible by using all TVs (i.e. 12). However, this solution is the upper bound, the lower bound is 4 TVs, this shows that there is a potential for possible improvement.

It is observed that most of the results have uniformity between QCs. This means that the number of TVs used for each QC is the same for one instance (i.e. all QCs use either 1, 2 or 3 TVs). This is mainly due to the fact that the number of positions loaded by each QC is the same. In one instance (240 container, 6 QCs, 200 minutes), the model finds a solution that requires one less TVs for one of the QCs compared the usage of remaining QCs.

## 6.7 Conclusion and future research direction

In this chapter, we have proposed a novel integrated container terminal problem. This problem focuses on the ship loading operations and tries to integrate the aspects of terminal-oriented stowage planning with routing and scheduling of transfer vehicles. We have formulated a mathematical model and a number of enhancements for this model. These enhancements can be used for similar research problems with some modifications. Results show that significant cost savings can be achieved with an efficient solution to this integrated problem rather than solving it in a hierarchical fashion. What is more, the enhancements that are suggested contribute to achieve a better solution. Results show that

we can load 60 containers in optimal way for most of the instances, while 240 containers are loaded with an average optimality gap of 21.1%.

We also test an alternative objective function for terminals. This problem aims at minimizing the number of handling equipment rather than minimizing the working time of these equipment. Results show that this problem is not necessarily an easier problem. Similar to original problem, we can load the 60-container instances with the optimal number of TV fleet. For loading 240 containers, the optimality gaps remain high for most of the instances.

We see many strong future research directions both on the problem definition and the solution method. With respect to the problem, we aim at going beyond some of the assumptions in the first place. The first clear addition would be integrating optimization of the load sequencing within the SLP. This extension will make the problem more complicated. However, the careful implementation of novel solution methods might obtain further cost savings. Another very promising research direction is to focus on the dual-cycling in which loading and unloading operations are executed simultaneously. Such an extension will allow a better utilization of the QCs and TVs.

Regarding the solution method, the lower bounding model which is relatively easy to solve can be used in a math-heuristic solution approach.

## APPENDIX A

# Appendix of Chapter 2

---

### A.1 Proof of Theorem 1

Assume that a better solution existed (resulting in fewer QC hours used). Let  $h_r$  be the number of hours we use  $r$  QCs in this solution, and  $h$  be the duration of the port stay ( $d(j)$  for given column  $j$ ). We must have:

$$\sum_{r \in \{r_{min}, \dots, r_{max}\}} h_r = h \quad (\text{A.1})$$

in order for the preemption constraint to be satisfied.

- (a) If  $h_r = 0$  for all  $r \in \{r_{min}, \dots, r_{max}\} \setminus \{q, q+1\}$  then we cannot do better than the solution computed in Algorithm (5) since here we computed all possible combinations by using  $q$  and  $q+1$  QCs.
- (b) Therefore, let us first analyze the situation where  $h_r = 0$  for all  $r < q$  and  $h_{\bar{r}} > 0$  for some  $\bar{r} > q+1$ . If  $h_q = 0$  then this solution is clearly worse than the one computed by the algorithm because of constraint (A.1). Thus we must have  $h_q > 0$  and we can make a more balanced solution that uses the same amount of QC hours by incrementing  $h_{q+1}$  and  $h_{\bar{r}-1}$  by one (if  $q+1 = \bar{r}-1$  then we increase  $h_{q+1}$  by 2) and decrementing  $h_q$  and  $h_{\bar{r}}$  by



one. We continue doing so until either  $h_q = 0$ , showing that the starting solution was not better than the one computed by the algorithm or  $h_q > 0$  and  $h_{\bar{r}} = 0$  for all  $\bar{r} > q + 1$ . Now we are back at case (a) and we see that the starting solution could not have used fewer QC hours than the one computed by the algorithm.

- (c) Now let us analyze the situation where  $h_r > 0$  for  $r < q$ . In that case we must have  $h_{\bar{r}} > 0$  for some  $\bar{r} > q$ . Otherwise we would have selected a lower  $q$  in the initial checks. We construct a more balanced solution that use the same number of QC hours by increasing  $h_{r+1}$  and  $h_{\bar{r}-1}$  by one (if  $r + 1 = \bar{r} - 1$  then this should be interpreted as increasing  $h_{r+1}$  by 2) and decreasing  $h_r$  and  $h_{\bar{r}}$  by one. By continuing to do so we either get to situation (a) or (b) and we see that the starting solution could not have used fewer QC hours than the one computed by the algorithm.  $\square$

## APPENDIX B

# Appendix of Chapter 3

---

### B.1 Calculation of $\hat{q}, f_1, f_2$ (Iris et al. (2015b))

---

**Algorithm 5:** QC assignment

---

**Input** :  $k, r_k^{min}, r_k^{max}, process_i, (1 + \beta\Delta b_i)m_k$

1 Find  $\hat{q} \in \{r_k^{min}, \dots, r_k^{max}\}$  such that  $\hat{q} = \left\lfloor \left( \frac{(1 + \beta\Delta b_i)m_k}{process_i} \right)^{1/\alpha} \right\rfloor$

2  $p = process_i$

3 **while** ( $p \geq 0$ ) **do**

4      $\delta = p(\hat{q} + 1)^\alpha + (process_i - p)(\hat{q})^\alpha$

5     **if**  $\delta \geq (1 + \beta\Delta b_i)m_k$  **then**

6          $f_2 = p, f_1 = (process_i - p)$

7      $p = p - 1$

8 **return**  $f_1, f_2, \hat{q}$

---

## B.2 Lower bound on objective function (Iris et al. (2015b))

We first obtain  $f_1, f_2, \hat{q}$  by using B.1. We know that an assignment  $j$  has a  $cost_j$  which is time-dependent cost component and QC assignment cost. With this information, we can calculate an improved lower bound  $\phi(j)$  for assignment  $j$ 's cost contribution to the objective function:

$$\phi(j) = cost_j + c_4(f_1\hat{q} + f_2(\hat{q} + 1))$$

We now use  $\phi(j)$  to define the lowest contribution  $\sigma_i$  for each vessel  $i$ :

$$\sigma_i = \min_{j \in \Omega_i} \{\phi(j)\}$$

and we compute the lower bound on the total objective:  $\underline{z}^1 = \sum_{i \in V} \sigma_i$ .

## B.3 Smarter greedy insertion: pseudo code

**Algorithm 6:** Smarter greedy insertion

---

**Definition:**  $I$ : Insertion list,  $X_p$ : partial solution,  $k$ : vessel index,  $i$ : assignment index,  $t$ : time index,  $\Omega_k$ : set of assignments of vessel  $k$  sorted in increasing cost order,  $i \in \Omega_k$

**Input** :  $I, X_p, \phi = |I|, p \in (0, 1], \Omega_k : \forall k \in I$

```

1 for  $k = I_1 \rightarrow I_\phi$  do
2    $i = 0, cost_{update} = \infty$ 
3   Draw a random number  $q$  in  $(0, 1]$ 
4   if  $(q > p)$  then
5      $i \rightarrow i + 1$ , go to 3
6   else
7     if  $(cost_{update} \leq cost_i)$  then
8        $X_p = i \cup X_p$ , apply  $QC_{use}^t$ , update Free_QC( $t$ )
9       Remove vessel  $k$  from insertion list:  $I = I - \{k\}$ 
10    else
11      if  $(fOverlap(i, X_p) = true) \wedge fCheckQCcapacity = enough)$ 
12        then
13           $X_p = i \cup X_p$ , Calculate  $\hat{q}, f_1, f_2$ , Make
14          QC_Assignment, update Free_QC( $t$ ),  $I = I - \{k\}$ 
15        else if  $(fOverlap(i, X_p) = true) \wedge fCheckQCcapacity = not enough)$  then
16           $QC_{needed} = (1 + \beta \Delta b_i) m_k$ 
17          for  $t = s_k \rightarrow s_k + process_i$  do
18            if  $(Free\_QC(t) < r_k^{min})$  then
19               $i \rightarrow i + 1$ , go to 3
20            else
21               $QC_{possible}^t = \min\{Free\_QC(t), r_k^{max}\}$ 
22               $QC_{effective} = QC_{effective} + (QC_{possible}^t)^\alpha$ 
23            if  $(QC_{effective} \geq QC_{needed})$  then
24              for  $t = s_k \rightarrow s_k + process_i$  do
25                 $QC_{use}^t = \max\{\min\{(\frac{QC_{needed}}{Left_{time}})^{1/\alpha}, QC_{possible}^t\}, r_k^{min}\}$ 
26                update Left_time,  $QC_{needed}$ ,  $cost_{update}$ 
27            else
28               $i \rightarrow i + 1$ , go to 3
29        else
30           $i \rightarrow i + 1$ , go to 3

```

---



# Bibliography

---

- A. Agra, M. Christiansen, A. Delgado, and L. M. Hvattum. A maritime inventory routing problem with stochastic sailing and port times. *Computers & Operations Research*, 61:18 – 30, 2015.
- N. Al-Dhaheri, A. Jebali, and A. Diabat. The quay crane scheduling problem with nonzero crane repositioning time and vessel stability constraints. *Computers & Industrial Engineering*, 94:230 – 244, 2016. ISSN 0360-8352.
- J. F. Alvarez. A heuristic for vessel planning in a reach stacker terminal. *Journal of Maritime Research*, 3(1):3–16, 2006.
- J. F. Alvarez. *Optimization Algorithms for Maritime Terminal and Fleet Management*. PhD thesis, Universitat Pompeu Fabra, 2008.
- J. F. Alvarez, T. Longva, and E. S. Engbrethsen. A methodology to assess vessel berthing and speed optimization policies. *Maritime Economics & Logistics*, 12:327–346, 2010.
- D. Ambrosino and A. Sciomachen. Impact of yard organisation on the master bay planning problem. *Maritime Economics & Logistics*, 5(3):285–300, 2003.
- D. Ambrosino, A. Sciomachen, and E. Tanfani. Stowing a containership: the master bay plan problem. *Transportation Research Part A: Policy and Practice*, 38(2):81 – 99, 2004. ISSN 0965-8564. doi: <http://dx.doi.org/10.1016/j.tra.2003.09.002>.
- D. Ambrosino, C. Caballini, and S. Siri. Mathematical model to evaluate different train loading and stacking policies. *Marit Econ Logist*, 15(3):292–308, Sep 2013. ISSN 1479-2931.

- E. Asperen, B. Borgman, and R. Dekker. Evaluating impact of truck announcements on container stacking efficiency. *Flexible Services and Manufacturing Journal*, 25(4):543–556, 2011. ISSN 1936-6590.
- Z. Bian, Q. Shao, and Z. Jin. Optimization on the container loading sequence based on hybrid dynamic programming. *Transport*, (ahead-of-print):1–10, 2015.
- C. Bierwirth and F. Meisel. A survey of berth allocation and quay crane scheduling problems in container terminals. *European Journal of Operational Research*, 202(3):615–627, May 2010. ISSN 03772217.
- C. Bierwirth and F. Meisel. A follow-up survey of berth allocation and quay crane scheduling problems in container terminals. *European Journal of Operational Research*, 000:1–15, Dec. 2014. ISSN 03772217.
- J. R. Birge and F. Louveaux. *Introduction to Stochastic Programming*. Springer Series in Operations Research and Financial Engineering. 2011. ISBN 978-1-4614-0236-7.
- E. Bish, F. Chen, Y. Leong, B. Nelson, J. Ng, and D. Simchi-Levi. Dispatching vehicles in a mega container terminal. *OR Spectrum*, 27(4):491–506, 2005. ISSN 0171-6468. doi: 10.1007/s00291-004-0194-2.
- E. K. Bish. A multiple-crane-constrained scheduling problem in a container terminal. *European Journal of Operational Research*, 144(1):83 – 107, 2003. ISSN 0377-2217.
- E. K. Bish, T.-Y. Leong, C.-L. Li, J. W. C. Ng, and D. Simchi-Levi. Analysis of a new vehicle scheduling and location problem. *Naval Research Logistics (NRL)*, 48(5):363–385, 2001. ISSN 1520-6750.
- J. Blazewicz, T. C. E. Cheng, M. Machowiak, and C. Oguz. Berth and quay crane allocation: a moldable task scheduling model. *Journal of the Operational Research Society*, 62(7):1189–1197, June 2010. ISSN 0160-5682.
- N. Boland, M. Fischetti, M. Monaci, and M. Savelsbergh. Proximity benders: a decomposition heuristic for stochastic programs. *Journal of Heuristics*, 22(2):181–198, 2015. ISSN 1572-9397.
- J. Bose, T. Reiners, D. Steenken, and S. Voß. Vehicle dispatching at seaport container terminals using evolutionary algorithms. In *System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on*, pages 10 pp. vol.2–, Jan 2000.
- J. W. Bose, editor. *Handbook of Terminal Planning*, volume 49. Springer New York, 2011. ISBN 978-1-4419-8407-4.

- B. Brinkmann. *Handbook of Terminal Planning*, chapter Operations Systems of Container Terminals: A Compendious Overview, pages 25–39. Springer New York, New York, NY, 2011. ISBN 978-1-4419-8408-1.
- K. Buhrkal, S. Zuglian, S. Ropke, J. Larsen, and R. Lusby. Models for the discrete berth allocation problem: A computational comparison. *Transportation Research Part E: Logistics and Transportation Review*, 47(4):461–473, July 2011. ISSN 13665545. doi: 10.1016/j.tre.2010.11.016.
- J. Cao, Q. Shi, and D.-H. Lee. Integrated quay crane and yard truck schedule problem in container terminals. *Tsinghua Science & Technology*, 15(4):467 – 474, 2010a. ISSN 1007-0214.
- J. X. Cao, D.-H. Lee, J. H. Chen, and Q. Shi. The integrated yard truck and yard crane scheduling problem: Benders’ decomposition-based methods. *Transportation Research Part E: Logistics and Transportation Review*, 46(3): 344 – 353, 2010b. ISSN 1366-5545.
- Y. Cardona-Valdes, A. Alvarez, and D. Ozdemir. A bi-objective supply chain design problem with uncertainty. *Transportation Research Part C: Emerging Technologies*, 19(5):821 – 832, 2011.
- H. J. Carlo, I. F. A. Vis, and K. J. Roodbergen. Seaside operations in container terminals: literature overview, trends, and research directions. *Flexible Services and Manufacturing Journal*, pages 1–39, 2013. ISSN 19366582. doi: 10.1007/s10696-013-9178-3.
- H. J. Carlo, I. F. A. Vis, and K. J. Roodbergen. Transport operations in container terminals: Literature overview, trends, research directions and classification scheme. *European Journal of Operational Research*, 236(1):1 – 13, 2014a.
- H. J. Carlo, I. F. A. Vis, and K. J. Roodbergen. Storage yard operations in container terminals: Literature overview, trends, and research directions. *European Journal of Operational Research*, 235(2):412 – 430, 2014b. Maritime Logistics.
- A. Carteni and S. de Luca. Tactical and strategic planning for a container terminal: Modelling issues within a discrete event simulation approach. *Simulation Modelling Practice and Theory*, 21(1):123 – 145, 2012.
- M. Caserta and S. Voß. A corridor method-based algorithm for the pre-marshalling problem. In M. Giacobini, A. Brabazon, S. Cagnoni, G. Di Caro, A. Ekárt, A. Esparcia-Alcázar, M. Farooq, A. Fink, and P. Machado, editors, *Applications of Evolutionary Computing*, volume 5484 of *Lecture Notes in Computer Science*, pages 788–797. Springer Berlin Heidelberg, 2009. ISBN 978-3-642-01128-3. doi: 10.1007/978-3-642-01129-0\_89.



- D. Chang, Z. Jiang, W. Yan, and J. He. Integrating berth allocation and quay crane assignments. *Transportation Research Part E: Logistics and Transportation Review*, 46(6):975–990, Nov. 2010. ISSN 13665545. doi: 10.1016/j.tre.2010.05.008.
- G. Chen, K. Govindan, and Z. Yang. Managing truck arrivals with time windows to alleviate gate congestion at container terminals. *International Journal of Production Economics*, 141(1):179 – 188, 2013a. Meta-heuristics for manufacturing scheduling and logistics problems.
- J. H. Chen, D.-H. Lee, and J. X. Cao. Heuristics for quay crane scheduling at indented berth. *Transportation Research Part E: Logistics and Transportation Review*, 47(6):1005 – 1020, 2011. ISSN 1366-5545.
- J. H. Chen, D.-H. Lee, and J. X. Cao. A combinatorial benders cuts algorithm for the quayside operation problem at container terminals. *Transportation Research Part E: Logistics and Transportation Review*, 48(1):266–275, Jan. 2012. ISSN 13665545. doi: 10.1016/j.tre.2011.06.004.
- L. Chen, A. Langevin, and Z. Lu. Integrated scheduling of crane handling and truck transportation in a maritime container terminal. *European Journal of Operational Research*, 225(1):142 – 152, 2013b. ISSN 0377-2217.
- T. Chen. Yard operations in the container terminal-a study in the ‘unproductive moves’. *Maritime Policy & Management*, 26(1):27–38, 1999. doi: 10.1080/030888399287041.
- R. K. Cheung, C.-L. Li, and W. Lin. Interblock crane deployment in container terminals. *Transportation Science*, 36(1):79–93, 2002.
- S. Choo, D. Klabjan, and D. Simchi-Levi. Multiship crane sequencing with yard congestion constraints. *Transportation Science*, 44(1):98–115, 2010.
- C. G. Christensen and C. T. Holst. Berth Allocation in Container Terminals (In Danish), 2008. URL <http://etd.dtu.dk/thesis/219860/>.
- J. Christensen, D. Pacino, J. Fonseca, and H. Psaraftis. Cargo-mix optimization in Liner Shipping. In *The 6th international workshop on Freight Transportation and Logistics (ODYSSEUS 2015)*, pages 41–45, 2015.
- G. Codato and M. Fischetti. Combinatorial benders’ cuts for mixed-integer linear programming. *Operations Research*, 54(4):756–766, 2006.
- L. C. Coelho, J. F. Cordeau, and G. Laporte. The inventory-routing problem with transshipment. *Computers & Operations Research*, 39:2537–2548, 2012. ISSN 03050548. doi: 10.1016/j.cor.2011.12.020.

- J.-F. Cordeau, G. Laporte, P. Legato, and L. Moccia. Models and Tabu Search Heuristics for the Berth-Allocation Problem. *Transportation Science*, 39(4): 526–538, Nov. 2005. ISSN 0041-1655. doi: 10.1287/trsc.1050.0120.
- J.-F. Cordeau, M. Gaudioso, G. Laporte, and L. Moccia. The service allocation problem at the gioia tauro maritime terminal. *European Journal of Operational Research*, 176(2):1167 – 1184, 2007. ISSN 0377-2217.
- J. F. Cordeau, G. Laporte, L. Moccia, and G. Sorrentino. Optimizing yard assignment in an automotive transshipment terminal. *European Journal of Operational Research*, 215:149–160, 2011. ISSN 03772217. doi: 10.1016/j.ejor.2011.06.008.
- J. Dai, W. Lin, R. Moorthy, and C.-P. Teo. *Supply Chain Analysis: A Handbook on the Interaction of Information, System and Optimization*, chapter Berth Allocation Planning Optimization in Container Terminals, pages 69–104. Springer US, Boston, MA, 2008.
- B. de Castillo and C. F. Daganzo. Handling strategies for import containers at marine terminals. *Transportation Research Part B: Methodological*, 27(2):151 – 166, 1993. ISSN 0191-2615.
- A. Diabat and E. Theodorou. An integrated quay crane assignment and scheduling problem. *Computers & Industrial Engineering*, 73:115 – 123, 2014. ISSN 0360-8352.
- Y. Du, Q. Chen, X. Quan, L. Long, and R. Y. Fung. Berth allocation considering fuel consumption and vessel emissions. *Transportation Research Part E: Logistics and Transportation Review*, 47(6):1021 – 1037, 2011. ISSN 1366-5545.
- G. Froyland, S. J. Maher, and C.-L. Wu. The recoverable robust tail assignment problem. *Transportation Science*, 48(3):351–372, 2014.
- Y.-M. Fu, A. Diabat, and I.-T. Tsai. A multi-vessel quay crane assignment and scheduling problem: Formulation and heuristic solution approach. *Expert Systems with Applications*, 41(15):6959 – 6965, 2014. ISSN 0957-4174.
- A. H. Gharehgozli, G. Laporte, Y. Yu, and R. de Koster. Scheduling twin yard cranes in a container block. *Transportation Science*, 49(3):686–705, 2015. doi: 10.1287/trsc.2014.0533.
- G. Giallombardo, L. Moccia, M. Salani, and I. Vacca. Modeling and solving the Tactical Berth Allocation Problem. *Transportation Research Part B: Methodological*, 44(2):232–245, Feb. 2010. ISSN 01912615. doi: 10.1016/j.trb.2009.07.003.

- M. Golias, G. K. Saharidis, M. Boile, S. Theofanis, and M. G. Ierapetruto. The berth allocation problem: Optimizing vessel arrival time. *Maritime Economics & Logistics*, 11:358–377, 2009.
- M. Golias, M. Boile, S. Theofanis, and C. Efstathiou. The berth-scheduling problem maximizing berth productivity and minimizing fuel consumption and emissions production. *Transportation Research Record: Journal of the Transportation Research Board*, 2166:20–27, 2010.
- M. Golias, I. Portal, D. Konur, E. Kaisar, and G. Kolomvos. Robust berth scheduling at marine container terminals via hierarchical optimization. *Computers & Operations Research*, 41:412 – 422, 2014. ISSN 0305-0548.
- A. V. Goodchild and C. F. Daganzo. Double-cycling strategies for container ships and their effect on ship loading and unloading operations. *Transportation Science*, 40(4):473–483, 2006.
- M. Grunow, H.-O. Günther, and M. Lehmann. Dispatching multi-load agvs in highly automated seaport container terminals. *OR Spectrum*, 26(2):211–235.
- M. Grunow, H.-O. Guenther, and M. Lehmann. Strategies for dispatching AGVs at automated seaport container terminals. *OR Spectrum*, 28(4):587–610, 2006. ISSN 0171-6468. doi: 10.1007/s00291-006-0054-3.
- Y. Han, L. H. Lee, E. P. Chew, and K. C. Tan. A yard storage strategy for minimizing traffic congestion in a marine container transshipment hub. *OR Spectrum*, 30(4):697–720, 2008. ISSN 1436-6304.
- P. Hansen, C. Oguz, and N. Mladenovic. Variable neighborhood search for minimum cost berth allocation. *European Journal of Operational Research*, 191(3):636 – 649, 2008. ISSN 0377-2217.
- J. He, Y. Huang, W. Yan, and S. Wang. Integrated internal truck, yard crane and quay crane scheduling in a container terminal considering energy consumption. *Expert Systems with Applications*, 42(5):2464 – 2487, 2015. ISSN 0957-4174.
- W. Hu, Z. Hu, L. Shi, P. Luo, and W. Song. Combinatorial optimization and strategy for ship stowage and loading schedule of container terminal. *Journal of computers*, 7(8):2078–2092, 2012.
- IBM. IBM ILOG CPLEX Optimization Studio V12.6.1 documentation. Technical report, 2015.
- A. Imai, E. Nishimura, and S. Papadimitriou. The dynamic berth allocation problem for a container port. *Transportation Research Part B: Methodological*, 35(4):401–417, May 2001. ISSN 01912615. doi: 10.1016/S0191-2615(99)00057-0.

- A. Imai, E. Nishimura, S. Papadimitriou, and K. Sasaki. The containership loading problem. *International Journal of Maritime Economics*, 4(2):126–148, 2002.
- A. Imai, E. Nishimura, and S. Papadimitriou. Berth allocation with service priority. *Transportation Research Part B: Methodological*, 37(5):437 – 457, 2003.
- A. Imai, X. Sun, E. Nishimura, and S. Papadimitriou. Berth allocation in a container port: using a continuous location space approach. *Transportation Research Part B: Methodological*, 39(3):199–221, Mar. 2005. ISSN 01912615. doi: 10.1016/j.trb.2004.04.004.
- A. Imai, K. Sasaki, E. Nishimura, and S. Papadimitriou. Multi-objective simultaneous stowage and load planning for a container ship with container rehandle in yard stacks. *European Journal of Operational Research*, 171(2): 373–389, 2006.
- A. Imai, E. Nishimura, M. Hattori, and S. Papadimitriou. Berth allocation at indented berths for mega-containerships. *European Journal of Operational Research*, 179(2):579 – 593, 2007. ISSN 0377-2217.
- A. Imai, H. C. Chen, E. Nishimura, and S. Papadimitriou. The simultaneous berth and quay crane allocation problem. *Transportation Research Part E: Logistics and Transportation Review*, 44(5):900–920, Sept. 2008. ISSN 13665545. doi: 10.1016/j.tre.2007.03.003.
- A. Imai, Y. Yamakawa, and K. Huang. The strategic berth template problem. *Transportation Research Part E: Logistics and Transportation Review*, 72:77–100, 2014. ISSN 13665545. doi: 10.1016/j.tre.2014.09.013.
- Ç. Iris and D. Pacino. A survey on the ship loading problem. In F. Corman, S. Voss, and R. R. Negenborn, editors, *Computational Logistics*, volume 9335 of *Lecture Notes in Computer Science*, pages 238–251. Springer International Publishing, 2015.
- Ç. Iris, D. Pacino, and S. Ropke. Improved formulations and an adaptive large neighborhood search heuristic for the integrated berth allocation and quay crane assignment problem. *Technical Report*, Technical University of Denmark, 2015a.
- Ç. Iris, D. Pacino, S. Ropke, and A. Larsen. Integrated berth allocation and quay crane assignment problem: Set partitioning models and computational results. *Transportation Research Part E: Logistics and Transportation Review*, 81:75 – 97, 2015b. ISSN 1366-5545. doi: <http://dx.doi.org/10.1016/j.tre.2015.06.008>.

- Ç. Iris, J. G. Jin, D. H. Lee, and S. Ropke. A two-stage stochastic programming approach to berth and quay crane scheduling problem under uncertainty. *Technical Report*, Technical University of Denmark, 2016a.
- Ç. Iris, D. Pacino, and S. Ropke. Formulations for ship loading problem with transport vehicle assignment and scheduling. *Technical Report*, Technical University of Denmark, 2016b.
- M. Ji, W. Guo, H. Zhu, and Y. Yang. Optimization of loading sequence and rehandling strategy for multi-quay crane operations in container terminals. *Transportation Research Part E: Logistics and Transportation Review*, 80(0): 1 – 19, 2015. ISSN 1366-5545.
- X. Jiang, L. H. Lee, E. P. Chew, Y. Han, and K. C. Tan. A container yard storage strategy for improving land utilization and operation efficiency in a transshipment hub port. *European Journal of Operational Research*, 221(1): 64 – 73, 2012. ISSN 0377-2217.
- X. Jiang, P. E. Chew, and H. L. Lee. *Handbook of Ocean Container Transport Logistics: Making Global Supply Chains Effective*, chapter Innovative Container Terminals to Improve Global Container Transport Chains, pages 3–41. Springer International Publishing, Cham, 2015. ISBN 978-3-319-11891-8.
- J. G. Jin, D.-H. Lee, and H. Hu. Tactical berth and yard template design at container transshipment terminals: A column generation based approach. *Transportation Research Part E: Logistics and Transportation Review*, 73: 168–184, 2015. ISSN 13665545. doi: 10.1016/j.tre.2014.11.009.
- J. G. Jin, D.-H. Lee, and J. X. Cao. Storage yard management in maritime container terminals. *Transportation Science*, Article-in-press(0):1–14, 2016. doi: 10.1287/trsc.2014.0527.
- R. Jovanovic and S. Voß. A chain heuristic for the blocks relocation problem. *Computers & Industrial Engineering*, 75:79 – 86, 2014.
- S. Jung and K. Kim. Load scheduling for multiple quay cranes in port container terminals. *Journal of Intelligent Manufacturing*, 17(4):479–492, 2006. ISSN 0956-5515. doi: 10.1007/s10845-005-0020-y.
- J. Kang, K. R. Ryu, and K. H. Kim. Deriving stacking strategies for export containers with uncertain weight information. *Journal of Intelligent Manufacturing*, 17(4):399–410, 2006. ISSN 1572-8145.
- J. Karafa, M. Golias, S. Ivey, G. Saharidis, and N. Leonardos. The berth allocation problem with stochastic vessel handling times. *The International Journal of Advanced Manufacturing Technology*, 65(1-4):473–484, 2013. ISSN 0268-3768. doi: 10.1007/s00170-012-4186-0.

- N. Kaveshgar and N. Huynh. Integrated quay crane and yard truck scheduling for unloading inbound containers. *International Journal of Production Economics*, 159:168 – 177, 2015. ISSN 0925-5273.
- N. Kemme. Container-terminal logistics. In *Design and Operation of Automated Container Storage Systems*, Contributions to Management Science, pages 9–52. Springer-Verlag Berlin Heidelberg, 2013.
- K. H. Kim and H. B. Kim. Segregating space allocation models for container inventories in port container terminals. *International Journal of Production Economics*, 59(1–3):415 – 423, 1999a. ISSN 0925-5273.
- K. H. Kim and K. Y. Kim. An optimal routing algorithm for a transfer crane in port container terminals. *Transportation Science*, 33(1):17–33, Jan. 1999b. ISSN 1526-5447. doi: 10.1287/trsc.33.1.17.
- K. H. Kim and K. Y. Kim. Routing straddle carriers for the loading operation of containers using a beam search algorithm. *Computers & Industrial Engineering*, 36(1):109 – 136, 1999c. ISSN 0360-8352.
- K. H. Kim and H. Lee. Container terminal operation: Current trends and future challenges. In C.-Y. Lee and Q. Meng, editors, *Handbook of Ocean Container Transport Logistics*, volume 220 of *International Series in Operations Research and Management Science*, pages 43–73. Springer International Publishing, 2015. ISBN 978-3-319-11890-1. doi: 10.1007/978-3-319-11891-8\_2.
- K. H. Kim and K. C. Moon. Berth scheduling by simulated annealing. *Transportation Research Part B: Methodological*, 37(6):541–560, July 2003. ISSN 01912615.
- K. H. Kim, Y. M. Park, and K.-R. Ryu. Deriving decision rules to locate export containers in container yards. *European Journal of Operational Research*, 124(1):89 – 101, 2000. ISSN 0377-2217.
- K. H. Kim, J. S. Kang, and K. R. Ryu. A beam search algorithm for the load sequencing of outbound containers in port container terminals. *OR Spectrum*, 26(1):93–116, 2004.
- K. Y. Kim and K. H. Kim. Heuristic algorithms for routing yard-side equipment for minimizing loading times in container terminals. *Naval Research Logistics (NRL)*, 50(5):498–514, 2003. ISSN 1520-6750. doi: 10.1002/nav.10076.
- F. Kiya and H. Davoudpour. Stochastic programming approach to re-designing a warehouse network under uncertainty. *Transportation Research Part E: Logistics and Transportation Review*, 48(5):919 – 936, 2012.
- E. Kozan and P. Preston. Mathematical modelling of container transfers and storage locations at seaport terminals. *OR Spectrum*, 28(4):519–537, 2006. ISSN 1436-6304.

- E. Lalla-Ruiz, J. L. González-Velarde, B. Melián-Batista, and J. M. Moreno-Vega. Biased random key genetic algorithm for the Tactical Berth Allocation Problem. *Applied Soft Computing Journal*, 22:60–76, 2014. ISSN 15684946.
- G. Laporte and F. V. Louveaux. The integer l-shaped method for stochastic integer programs with complete recourse. *Operations Research Letters*, 13(3): 133 – 142, 1993. ISSN 0167-6377.
- H. Lau and Y. Zhao. Integrated scheduling of handling equipment at automated container terminals. *Annals of Operations Research*, 159(1):373–394, 2008. ISSN 0254-5330. doi: 10.1007/s10479-007-0262-5.
- X. le Han, Z. qiang Lu, and L. feng Xi. A proactive approach for simultaneous berth and quay crane scheduling problem with stochastic arrival and handling time. *European Journal of Operational Research*, 207(3):1327 – 1340, 2010. ISSN 0377-2217.
- D.-H. Lee and J. G. Jin. Feeder vessel management at container transshipment terminals. *Transportation Research Part E: Logistics and Transportation Review*, 49(1):201 – 216, 2013. ISSN 1366-5545.
- D.-H. Lee, Z. Cao, and Q. Meng. Scheduling of two-transtainer systems for loading outbound containers in port container terminals with simulated annealing algorithm. *International Journal of Production Economics*, 107(1):115 – 124, 2007. ISSN 0925-5273. doi: <http://dx.doi.org/10.1016/j.ijpe.2006.08.003>. Special Section on Building Core-Competence through Operational Excellence.
- D.-H. Lee, J. X. Cao, Q. Shi, and J. H. Chen. A heuristic algorithm for yard truck scheduling and storage allocation problems. *Transportation Research Part E: Logistics and Transportation Review*, 45(5):810 – 820, 2009a. ISSN 1366-5545.
- D.-H. Lee, J. X. Cao, and Q. X. Shi. Synchronization of yard truck scheduling and storage allocation in container terminals. *Engineering Optimization*, 41 (7):659–672, 2009b.
- D. H. Lee, J. H. Chen, and J. X. Cao. The continuous Berth Allocation Problem: A Greedy Randomized Adaptive Search Solution. *Transportation Research Part E: Logistics and Transportation Review*, 46(6):1017–1029, 2010a. ISSN 13665545.
- D.-H. Lee, J. G. Jin, and C. Jiang. Integrated bay allocation and yard crane scheduling problem for transshipment containers. *Journal of the Transportation Research Board*, 2222:63–71, 2011. doi: 10.3141/2222-08.

- D.-H. Lee, J. G. Jin, and J. H. Chen. Terminal and yard allocation problem for a container transshipment hub with multiple terminals. *Transportation Research Part E: Logistics and Transportation Review*, 48(2):516 – 528, 2012.
- L. Lee, E. Chew, K. Tan, and Y. Wang. Vehicle dispatching algorithms for container transshipment hubs. *OR Spectrum*, 32(3):663–685, 2010b. ISSN 0171-6468. doi: 10.1007/s00291-010-0199-y.
- L. H. Lee, E. P. Chew, K. C. Tan, and Y. Han. An optimization model for storage yard management in transshipment hubs. *OR Spectrum*, 28(4):539–561, 2006. ISSN 1436-6304.
- Y. Lee and N.-Y. Hsu. An optimization model for the container pre-marshalling problem. *Computers & Operations Research*, 34(11):3295 – 3313, 2007.
- Y. H. Lee, J. Kang, K. R. Ryu, and K. H. Kim. Optimization of container load sequencing by a hybrid of ant colony optimization and tabu search. In Y. S. O. Lipo Wang, Ke Chen, editor, *Advances in Natural Computation*, pages 1259–1268. Springer, 2005.
- P. Legato and R. M. Mazza. Managing container reshuffling in vessel loading by simulation. In *Proceedings of the 2013 Winter Simulation Conference: Simulation: Making Decisions in a Complex World*, pages 3450–3461. IEEE Press, 2013.
- P. Legato, D. Gulli, and R. Trunfio. The quay crane deployment problem at a maritime container terminal. In *Proceedings 22nd European Conference on Modelling and Simulation*, 2008. doi: 10.7148/2008-0053.
- P. Legato, R. Mazza, and R. Trunfio. Simulation-based optimization for discharge/loading operations at a maritime container terminal. *OR Spectrum*, 32(3):543–567, 2010. ISSN 0171-6468. doi: 10.1007/s00291-010-0207-2.
- P. Legato, R. Trunfio, and F. Meisel. Modeling and solving rich quay crane scheduling problems. *Computers & Operations Research*, 39(9):2063 – 2078, 2012. ISSN 0305-0548.
- J. Lehnfeld and S. Knust. Loading, unloading and premarshalling of stacks in storage areas: Survey and classification. *European Journal of Operational Research*, 239(2):297 – 312, 2014. ISSN 0377-2217.
- M. Levinson. *The Box: How the Shipping Container Made the World Smaller and the World Economy Bigger*. Princeton University Press, 2008. ISBN 9781400828586.
- C.-L. Li and G. L. Vairaktarakis. Loading and unloading operations in container terminals. *IIE Transactions*, 36(4):287–297, 2004.



- F. Li, J.-B. Sheu, and Z.-Y. Gao. Solving the continuous berth allocation and specific quay crane assignment problems with quay crane coverage range. *Transportation Science*, 49(4):968–989, 2015a.
- M. Z. Li, J. G. Jin, and C. X. Lu. Real-time disruption recovery for integrated berth allocation and crane assignment in container terminals. *Transportation Research Record: Journal of the Transportation Research Board*, 2479:49–59, 2015b.
- W. Li, M. Goh, Y. Wu, M. Petering, R. de Souza, and Y. Wu. A continuous time model for multiple yard crane scheduling with last minute job arrivals. *International Journal of Production Economics*, 136(2):332 – 343, 2012. ISSN 0925-5273.
- C. Liang, Y. Huang, and Y. Yang. A quay crane dynamic scheduling problem by hybrid evolutionary algorithm for berth allocation planning. *Computers & Industrial Engineering*, 56(3):1021–1028, Apr. 2009. ISSN 03608352. doi: 10.1016/j.cie.2008.09.024.
- J. Liu, Y.-w. Wan, and L. Wang. Quay crane scheduling at container terminals to minimize the maximum relative tardiness of vessel departures. *Naval Research Logistics*, 53(1):60–74, Feb. 2006. ISSN 0894-069X. doi: 10.1002/nav.20108.
- J. Luo, Y. Wu, A. Halldorsson, and X. Song. Storage and stacking logistics problems in container terminals. *OR Insight*, 24:256–275, 2011. ISSN 0953-5543.
- S. J. Maher, G. Desaulniers, and F. Soumis. Recoverable robust single day aircraft maintenance routing problem. *Computers & Operations Research*, 51:130 – 145, 2014.
- F. Meisel. Seaside Operations Planning in Container Terminals. 2009a. doi: 10.1007/978-3-7908-2191-8.
- F. Meisel. *Seaside Operations Planning in Container Terminals*, chapter Maritime Container Transport, pages 5–15. Physica-Verlag HD, Heidelberg, 2009b. ISBN 978-3-7908-2191-8.
- F. Meisel. *Seaside Operations Planning in Container Terminals*, chapter Operational Planning Problems, pages 17–30. Physica-Verlag HD, Heidelberg, 2009c. ISBN 978-3-7908-2191-8.
- F. Meisel. *Seaside Operations Planning in Container Terminals*, chapter Integration Concepts for Seaside Operations Planning, pages 47–54. Physica-Verlag HD, Heidelberg, 2009d. ISBN 978-3-7908-2191-8.
- F. Meisel. *Seaside Operations Planning in Container Terminals*, chapter Berth Allocation and Quay Crane Assignment, pages 55–83. Physica-Verlag HD, Heidelberg, 2009e. ISBN 978-3-7908-2191-8.

- F. Meisel. The quay crane scheduling problem with time windows. *Naval Research Logistics (NRL)*, 58(7):619–636, 2011. ISSN 1520-6750.
- F. Meisel and C. Bierwirth. Heuristics for the integration of crane productivity in the berth allocation problem. *Transportation Research Part E: Logistics and Transportation Review*, 45(1):196–209, Jan. 2009. ISSN 13665545. doi: 10.1016/j.tre.2008.03.001.
- F. Meisel and C. Bierwirth. A unified approach for the evaluation of quay crane scheduling models and algorithms. *Computers & Operations Research*, 38(3): 683 – 693, 2011. ISSN 0305-0548. doi: <http://dx.doi.org/10.1016/j.cor.2010.08.001>.
- F. Meisel and C. Bierwirth. A Framework for Integrated Berth Allocation and Crane Operations Planning in Seaport Container Terminals. *Transportation Science*, 47(2):131–147, May 2013. ISSN 0041-1655. doi: 10.1287/trsc.1120.0419.
- F. Meisel and M. Wichmann. Container sequencing for quay cranes with internal reshuffles. *OR spectrum*, 32(3):569–591, 2010.
- A. Minsaa and H. N. Psaraftis. *Green Transportation Logistics: The Quest for Win-Win Solutions*, chapter Directions for Further Research, pages 517–542. Springer International Publishing, Cham, 2016. ISBN 978-3-319-17175-3.
- L. Moccia, J.-F. Cordeau, M. Gaudioso, and G. Laporte. A branch-and-cut algorithm for the quay crane scheduling problem in a container terminal. *Naval Research Logistics (NRL)*, 53(1):45–59, 2006. ISSN 1520-6750.
- L. Moccia, J.-F. Cordeau, M. F. Monaco, and M. Sammarra. A column generation heuristic for a dynamic generalized assignment problem. *Computers & Operations Research*, 36(9):2670 – 2681, 2009. ISSN 0305-0548.
- F. Monaco and M. Sammarra. The berth allocation problem: A strong formulation solved by a lagrangean approach. *Transportation Science*, 41(2):265–280, 2007. doi: 10.1287/trsc.1060.0171.
- M. F. Monaco, M. Sammarra, and G. Sorrentino. The terminal-oriented ship stowage planning problem. *European Journal of Operational Research*, 239 (1):256–265, 2014.
- R. Moorthy and C.-P. Teo. Berth management in container terminal: the template design problem. *OR Spectrum*, 28(4):495–518, 2006. ISSN 0171-6468. doi: 10.1007/s00291-006-0036-5.
- L. F. Muller, S. Spoorendonk, and D. Pisinger. A hybrid adaptive large neighborhood search heuristic for lot-sizing with setup times. *European Journal of Operational Research*, 218(3):614 – 623, 2012. ISSN 0377-2217.

- W. Ng and K. Mak. Yard crane scheduling in port container terminals. *Applied Mathematical Modelling*, 29(3):263 – 276, 2005. ISSN 0307-904X.
- E. Nishimura, A. Imai, and S. Papadimitriou. Yard trailer routing at a maritime container terminal. *Transportation Research Part E: Logistics and Transportation Review*, 41(1):53 – 76, 2005.
- T. E. Notteboom. The time factor in liner shipping services. *Maritime Economics and Logistics*, 8:19 – 39, 2006.
- N. Noyan. Risk-averse two-stage stochastic programming with an application to disaster management. *Computers & Operations Research*, 39(3):541 – 559, 2012.
- F. Oliveira, P. Nunes, R. Blajberg, and S. Hamacher. A framework for crude oil scheduling in an integrated terminal-refinery system under supply uncertainty. *European Journal of Operational Research*, pages –, 2016.
- D. Pacino. *Fast Generation of Container Vessel Stowage Plans*. PhD thesis, IT University of Copenhagen, Denmark, 2012.
- D. Pacino, A. Delgado, R. M. Jensen, and T. Bebbington. Fast generation of near-optimal plans for eco-efficient stowage of large container vessels. In *Computational Logistics*, pages 286–301. Springer, 2011.
- K.-W. Pang and J. Liu. An integrated model for ship routing with transshipment and berth allocation. *IIE Transactions*, 46(12):1357–1370, 2014.
- K.-W. Pang, Z. Xu, and C.-L. Li. Ship routing problem with berthing time clash avoidance constraints. *International Journal of Production Economics*, 131(2):752 – 762, 2011. ISSN 0925-5273.
- T. Park, R. Choe, Y. H. Kim, and K. R. Ryu. Dynamic adjustment of container stacking policy in an automated container terminal. *International Journal of Production Economics*, 133(1):385 – 392, 2011. ISSN 0925-5273. Leading Edge of Inventory Research.
- Y.-M. Park and K. H. Kim. A scheduling method for Berth and Quay cranes. *OR Spectrum*, 25(1):1–23, Feb. 2003. ISSN 0171-6468. doi: 10.1007/s00291-002-0109-z.
- M.-H. Phan and K. H. Kim. Negotiating truck arrival times among trucking companies and a container terminal. *Transportation Research Part E: Logistics and Transportation Review*, 75:132 – 144, 2015. ISSN 1366-5545.
- M.-H. Phan and K. H. Kim. Collaborative truck scheduling and appointments for trucking companies and container terminals. *Transportation Research Part B: Methodological*, 86:37 – 50, 2016. ISSN 0191-2615.

- M. Pishvaei, J. Razmi, and S. Torabi. An accelerated benders decomposition algorithm for sustainable supply chain network design under uncertainty: A case study of medical needle and syringe supply chain. *Transportation Research Part E: Logistics and Transportation Review*, 67:14 – 38, 2014. ISSN 1366-5545.
- D. Pisinger and S. Ropke. A general heuristic for vehicle routing problems. *Computers & Operations Research*, 34(8):2403 – 2435, 2007. ISSN 0305-0548. doi: <http://dx.doi.org/10.1016/j.cor.2005.09.012>.
- H. N. Psaraftis and C. A. Kontovas. Balancing the economic and environmental performance of maritime transportation. *Transportation Research Part D: Transport and Environment*, 15(8):458 – 462, 2010.
- B. Raa, W. Dullaert, and R. V. Schaeren. An enriched model for the integrated berth allocation and quay crane assignment problem. *Expert Systems with Applications*, 38(11):14136–14147, May 2011. ISSN 09574174. doi: 10.1016/j.eswa.2011.04.224.
- W. Rei, M. Gendreau, and P. Soriano. Local branching cuts for the 0-1 integer l-shaped algorithm. *Technical Report*, CIRRELT-2007-23, 2007.
- L. B. Reinhardt, C. E. Plum, D. Pisinger, M. M. Sigurd, and G. T. Vial. The liner shipping berth scheduling problem with transit times. *Transportation Research Part E: Logistics and Transportation Review*, 86:116 – 128, 2016. ISSN 1366-5545.
- N. J. Rezanova and D. M. Ryan. The train driver recovery problem—a set partitioning based model and solution method. *Computers & Operations Research*, 37(5):845 – 856, 2010. ISSN 0305-0548. Disruption Management.
- T. Robenek, N. Umang, M. Bierlaire, and S. Ropke. A branch-and-price algorithm to solve the integrated berth allocation and yard assignment problem in bulk ports. *European Journal of Operational Research*, 235(2):399–411, 2014. ISSN 03772217. doi: 10.1016/j.ejor.2013.08.015.
- J.-P. Rodrigue. *The Geography of Transport Systems*, volume 3. New York: Routledge, 2013. ISBN 978-0-415-82254-1.
- M. Rodriguez-Molins, L. Ingolotti, F. Barber, M. Salido, M. Sierra, and J. Puente. A genetic algorithm for robust berth allocation and quay crane assignment. *Progress in Artificial Intelligence*, 2(4):177–192, 2014a. ISSN 2192-6352. doi: 10.1007/s13748-014-0056-3.
- M. Rodriguez-Molins, M. a. Salido, and F. Barber. A GRASP-based meta-heuristic for the Berth Allocation Problem and the Quay Crane Assignment Problem by managing vessel cargo holds. *Applied Intelligence*, 40:273–290, 2014b. ISSN 0924669X.

- S. Ropke and D. Pisinger. An Adaptive Large Neighborhood Search Heuristic for the Pickup and Delivery Problem with Time Windows. *Transportation Science*, 40(December 2014):455–472, 2006. ISSN 0041-1655. doi: 10.1287/trsc.1050.0135.
- Y. Saadaoui, N. Umang, and E. Frejinger. A Column Generation Framework for Berth Scheduling at Port Terminals. Technical report, 2015. URL <https://www.cirrelt.ca/DocumentsTravail/CIRRELT-2015-15.pdf>.
- T. Santoso, S. Ahmed, M. Goetschalckx, and A. Shapiro. A stochastic programming approach for supply chain network design under uncertainty. *European Journal of Operational Research*, 167(1):96 – 115, 2005. ISSN 0377-2217.
- S. Sauri and E. Martin. Space allocating strategies for improving import yard performance at marine terminals. *Transportation Research Part E: Logistics and Transportation Review*, 47(6):1038 – 1057, 2011. ISSN 1366-5545.
- A. Sciomachen and E. Tanfani. A 3d-bpp approach for optimising stowage plans and terminal productivity. *European Journal of Operational Research*, 183(3): 1433–1446, 2007.
- P. Shaw. Using constraint programming and local search methods to solve vehicle routing problems. *Principles and Practice of Constraint Programming - Cp98*, 1520:417–431, 1998. ISSN 0302-9743. doi: 10.1007/3-540-49481-2\\_30.
- K. Shin and T. Lee. Container loading and unloading scheduling for a mobile harbor system: a global and local search method. *Flexible Services and Manufacturing Journal*, 25(4):557–575, 2012. ISSN 1936-6590.
- R. Stahlbock and S. Voß. *The Vehicle Routing Problem: Latest Advances and New Challenges*, chapter Vehicle Routing Problems and Container Terminal Operations – An Update of Research, pages 551–589. Springer US, Boston, MA, 2008. ISBN 978-0-387-77778-8.
- R. Stahlbock and S. Voß. Operations research at container terminals: a literature update. *OR Spectrum*, 30(1):1–52, Oct. 2007. ISSN 0171-6468. doi: 10.1007/s00291-007-0100-9.
- D. Steenken, T. Winter, and U. T. Zimmermann. Stowage and transport optimization in ship planning. In J. R. Martin Groetschel, Sven O. Krumke, editor, *Online optimization of large scale systems*, pages 731–745. Springer, 2001.
- D. Steenken, S. Voß, and R. Stahlbock. Container terminal operation and operations research - A classification and literature review. *OR Spectrum*, 26: 3–49, 2004. ISSN 01716468. doi: 10.1007/3-540-26686-0\\_1.

- L. Tang, J. Zhao, and J. Liu. Modeling and solution of the joint quay crane and truck scheduling problem. *European Journal of Operational Research*, 236(3): 978 – 990, 2014. ISSN 0377-2217. Vehicle Routing and Distribution Logistics.
- E. Theodorou and A. Diabat. A joint quay crane assignment and scheduling problem: formulation, solution algorithm and computational results. *Optimization Letters*, 9(4):799–817, 2014. ISSN 1862-4480.
- S. Trukhanov, L. Ntaimo, and A. Schaefer. Adaptive multicut aggregation for two-stage stochastic linear programs with recourse. *European Journal of Operational Research*, 206(2):395 – 406, 2010. ISSN 0377-2217.
- Y. B. Turkogulari, Z. C. Taskin, N. Aras, and I. K. Altinel. Optimal berth allocation, time-variant quay crane assignment and scheduling with crane setups in container terminals. *European Journal of Operational Research*, Articles-in-advance, 2016. doi: <http://dx.doi.org/10.1016/j.ejor.2016.04.022>. URL [http://www.ie.boun.edu.tr/~taskin/pdf/bacasp\\_timevarying.pdf](http://www.ie.boun.edu.tr/~taskin/pdf/bacasp_timevarying.pdf).
- Y. B. Turkogullari, Z. C. Taskin, N. Aras, and I. K. Altinel. Optimal berth allocation and time-invariant quay crane assignment in container terminals. *European Journal of Operational Research*, 235(1):88–101, May 2014. ISSN 03772217. doi: 10.1016/j.ejor.2013.10.015.
- N. Umang, M. Bierlaire, and I. Vacca. Exact and heuristic methods to solve the berth allocation problem in bulk ports. *Transportation Research Part E: Logistics and Transportation Review*, 54:14–31, Aug. 2013. ISSN 13665545. doi: 10.1016/j.tre.2013.03.003.
- N. Umang, M. Bierlaire, and A. L. Erera. Real-time management of berth allocation with stochastic arrival and handling times. *Journal of Scheduling*, In-print, 2016.
- UNCTAD. *Review of Maritime Transport: 2013*. 2014. ISBN 9789211128109.
- UNCTAD. Review of maritime transport 2014. In *United Nations Conference on Trade and Development*, 2015.
- O. Unsal and C. Oguz. Constraint programming approach to quay crane scheduling problem. *Transportation Research Part E: Logistics and Transportation Review*, 59:108 – 122, 2013. ISSN 1366-5545.
- E. Ursavas. A decision support system for quayside operations in a container terminal. *Decision Support Systems*, 59:312–324, 2014. ISSN 01679236. doi: 10.1016/j.dss.2014.01.003.
- E. Ursavas. Crane allocation with stability considerations. *Maritime Economics and Logistics*, pages –, 2016. ISSN 1479-2931. URL <http://dx.doi.org/10.1057/me1.2015.35>.

- I. Vacca, M. Salani, and M. Bierlaire. An Exact Algorithm for the Integrated Planning of Berth Allocation and Quay Crane Assignment. *Transportation Science*, 47(2):148–161, May 2013. ISSN 0041-1655. doi: 10.1287/trsc.1120.0428.
- G. Venturini, Ç. Iris, and A. Larsen. Multiple-port berth allocation and speed optimization problem: models and computational results. Technical report, 2016.
- Y. Vergados, R. V. Schaeren, W. Dullaert, and B. Raa. The Berth Allocation and Quay Crane Assignment Problem Using a CP Approach. In *Principles and Practice of Constraint Programming*, pages 880–896. Lecture Notes in Computer Science, 2013.
- I. F. A. Vis. Survey of research in the design and control of automated guided vehicle systems. *European Journal of Operational Research*, 170(3):677 – 709, 2006a.
- I. F. A. Vis. A comparative analysis of storage and retrieval equipment at a container terminal. *International Journal of Production Economics*, 103(2): 680 – 693, 2006b.
- I. F. A. Vis and R. G. Anholt. Performance analysis of berth configurations at container terminals. *OR Spectrum*, 32(3):453–476, 2010. ISSN 1436-6304.
- I. F. A. Vis and R. de Koster. Transshipment of containers at a container terminal: An overview. *European Journal of Operational Research*, 147(1):1 – 16, 2003. ISSN 0377-2217.
- I. F. A. Vis, R. M. B. M. de Koster, K. J. Roodbergen, and L. Peeters. Determination of the number of automated guided vehicles required at a semi-automated container terminal. *The Journal of the Operational Research Society*, 52(4):409–417, 2001.
- I. F. A. Vis, R. M. B. M. de Koster, and M. W. P. Savelsbergh. Minimum vehicle fleet size under time-window constraints at a container terminal. *Transportation Science*, 39(2):249–260, 2005. doi: 10.1287/trsc.1030.0063.
- S. Wang, Q. Meng, and Z. Liu. A note on “berth allocation considering fuel consumption and vessel emissions”. *Transportation Research Part E: Logistics and Transportation Review*, 49(1):48 – 54, 2013. ISSN 1366-5545.
- S. Wang, Z. Liu, and X. Qu. Collaborative mechanisms for berth allocation. *Advanced Engineering Informatics*, 29(3):332 – 338, 2015. ISSN 1474-0346.
- Y. Wang and K. H. Kim. A quay crane scheduling algorithm considering the workload of yard cranes in a container yard. *Journal of Intelligent Manufacturing*, 22(3):459–470, 2009. ISSN 1572-8145.

- J. Wiese, L. Suhl, and N. Kliewer. *Handbook of Terminal Planning*, chapter Planning Container Terminal Layouts Considering Equipment Types and Storage Block Design, pages 219–245. Springer New York, New York, NY, 2011. ISBN 978-1-4419-8408-1.
- S. H. Won, X. Zhang, and K. H. Kim. Workload-based yard-planning system in container terminals. *Journal of Intelligent Manufacturing*, 23(6):2193–2206, 2011. ISSN 1572-8145.
- Y. Wu, W. Li, M. E. H. Petering, M. Goh, and R. de Souza. Scheduling multiple yard cranes with crane interference and safety distance requirement. *Transportation Science*, 49(4):990–1005, 2015.
- C. Yang, X. Wang, and Z. Li. An optimization approach for coupling problem of berth allocation and quay crane assignment in container terminal. *Computers & Industrial Engineering*, 63(1):243–253, Aug. 2012. ISSN 03608352. doi: 10.1016/j.cie.2012.03.004.
- F. You and I. E. Grossmann. Multicut benders decomposition algorithm for process supply chain planning under uncertainty. *Annals of Operations Research*, 210(1):191–211, 2011. ISSN 1572-9338.
- C. Zhang, Y. wah Wan, J. Liu, and R. J. Linn. Dynamic crane deployment in container storage yards. *Transportation Research Part B: Methodological*, 36(6):537 – 555, 2002. ISSN 0191-2615.
- C. Zhang, J. Liu, Y. wah Wan, K. G. Murty, and R. J. Linn. Storage space allocation in container terminals. *Transportation Research Part B: Methodological*, 37(10):883 – 903, 2003. ISSN 0191-2615.
- N. Zhao, W. Mi, C. Mi, and J. Chai. Study on vessel slot planning problem in stowage process of outbound containers. *Journal of Applied Sciences*, 13: 4278–4285, 2013.
- L. Zhen. Tactical berth allocation under uncertainty. *European Journal of Operational Research*, 247(3):928 – 944, 2015. ISSN 0377-2217.
- L. Zhen, E. P. Chew, and L. H. Lee. An integrated model for berth template and yard template planning in transshipment hubs. *Transportation Science*, 45(4):483–504, 2011a.
- L. Zhen, L. H. Lee, and E. P. Chew. A decision model for berth allocation under uncertainty. *European Journal of Operational Research*, 212(1):54 – 68, 2011b. ISSN 0377-2217.